Information Technology Grade 12

Lesson Notes and Worksheets

Foreword

In 1953 Nelson Mandela said, "...you must make every home, every shack, every rickety structure a centre of learning for our children."

Mindset is a South African based non-profit organisation, launched in 2003 by Nelson Mandela and created to do just what Mr Mandela requested. Mindset develops and distributes high quality, contextually relevant educational content for use in the schooling and health sectors in video, print and computer-based multimedia formats. School-related material is developed through Mindset Learn.

Mindset Learn seeks primarily to support teachers and learners; teachers through the provision of content and training to enhance and improve classroom practice, and learners by providing content both to support their understanding of the curriculum and also to act as additional tuition support. Mindset Learn has developed comprehensive support for FET Mathematics, Physical Sciences, Life Sciences, Mathematical Literacy and English First Additional Language. This Grade 12 Information Technology material now concludes our development of comprehensive support for FET IT.

The Grade 12 IT material has been designed explicitly to make the learning and teaching of IT exciting, impactful and relevant. In it, teachers will find expertly designed and produced resources to support them in bringing the curriculum to life and engaging their learners as they develop the skills and understanding of IT required by the curriculum. Teachers will also find numerous innovative ideas for the use of these resources in their classrooms.

At Mindset, we believe that quality education is one of the bedrocks of a healthy, prosperous and harmonious society. We also believe that teachers are the essential drivers of quality education. No amount of resourcing or technology will ever replace the impact and effectiveness of a passionate and dedicated teacher. It is our hope that this material will inspire and motivate teachers across South Africa to continue to provide the best quality education they can for the learners under their care and that it will impact positively on their professional lives.

Thanks must go to the Mindset content development and production teams and the host of contributing teachers for their tireless work in creating resources to the highest educational and technical standards possible.

Finally, an enormous debt of gratitude is owed to Mindset's partner and funder of this material, CoZa Cares. Without their continued and unwavering support for education in South Africa, none of this would be possible.

D**ylan Busa**
Mindset Learn Executive

Contents

Introduction

Purpose of the material

This print module has been written to support the use of the five series of Mindset Learn video lessons for Grade 12 Information Technology. The series are: Database Applications, Programming in Delphi and Java, and The Practical Assessment Task. Together these address all the Learning Outcomes and assessment standards in the National Curriculum Statements for Grade 12 learners.

How the material is organised

This module provides a teacher guide for each series. It also contains lesson notes and task answers for each of the series of videos, as well as worksheets, with tasks, and worksheet/task answers for each of the series of interactive, computer-based media.

An overview of the series

Database Applications (six interactive, computer-based lessons)

- File Types and Data Storage
- Database Tables
- Entity Relationship Modelling
- Normalising Databases
- Database Queries
- Database Maintenance

Programming in Delhi and Java (twenty interactive, computer-based lessons, covering the two programming languages simultaneously)

- Simple Data Types
- Control Structures: Selection
- Control Structures: Sequence
- Control Structures: Repetition
- Task Analysis I
- Task Analysis II
- Functions and Procedures I
- Functions and Procedures II
- String Manipulation
- Arrays I
- Arrays II
- Arrays III
- Classes and Objects I
- Classes and Objects II
- Classes and Objects III
- Text Files
- Evaluating
- Graphic User Interface I
- Graphic User Interface II
- SQL

The Practical Assessment Task (twelve video lessons covering phases 1 and 2, four interactive, computer-based lessons for phase 3)

The PAT Video lessons

- Phase 1 - Investigation and Analysis of a Problem
- Phase 1 - Identifying the Requirements

- Phase 2 - Program Specifications
- Phase 2 - Algorithms I
- Phase 2 - Algorithms II
- Phase 2 - Drawing up IPO Tables
- Phase 2 - Storing Data in IPO Tables
- Phase 2 - Classes, Objects and Data Flow I
- Phase 2 - Classes, Objects and Data Flow II
- Phase 2 - Database Design
- Phase 2 - Database Design - Final Requirements
- Phase 2 - Graphic User Interface

The PAT Interactive, computer-based lessons

- Phase 3 - Database Tables
- Phase 3 - Database Relationships
- Phase 3 - GUI Components
- Phase 3 - The GUI and the User

The Social and Ethical Impact of ICTs (three video lessons)

- The Impact of Information and Communication Technologies on our Lives
- How Technology can Benefit or Harm the User, the Environment and Society
- Cyber Threats and Safeguards to Protect you against Attacks

Troubleshooting Hardware and System Software Problems (five video lessons)

- Troubleshooting inside your computer
- Troubleshooting components and devices
- Troubleshooting Hardware and Systems Software
- Troubleshooting Networks
- Troubleshooting the Internet

The teacher guide

The teacher guide to each series is intended to help teachers plan how to use the video or interactive, computer-based lessons in their learning programme.

The teacher guide:

- gives an overview of the content of each lesson in the series, and the lesson outcomes that each lesson will help learners achieve
- shows how the lessons link to the Learning Outcomes and Assessment Standards (ASs) of the National Curriculum Statement (NCS)
- describes and explains the teaching approach taken in the videos and interactive, computer-based lessons
- offers suggestions for how teachers can integrate the videos and interactive, computer-based lessons into their classroom
- suggests ways to make good use of the tasks set in each video or interactive, computer-based lessons

The lesson notes

There is a lesson note for each video in three of the five series. Each note:

- provides a summary of the main learning points made in the video
- reproduces some of the key graphics shown

- gives the lesson outcomes for the lesson, and the Learning Outcome/s and Assessment Standards to which these are linked
- reproduces the task which is designed to assess learners' achievement of the outcomes

The lesson notes are intended for learners as well as teachers. They can be copied and given to learners to provide them with a good record of the video. This will encourage them to watch carefully, without trying to write everything down. The lesson notes can be used to supplement textbooks and other materials used in the classroom, and are useful even if learners do not watch all the video lessons in a series. These notes also record the task for each video lesson.

The worksheets

There is a worksheet for each interactive, computer-based lesson, in three of the five series. Each worksheet:

- provides a series of exercises for completion while working through an interactive, computer-based lesson
- offers a mechanism to ensure active engagement with the learning material while working with the lesson
- provides an opportunity for ongoing assessment, if required
- also includes a task based on the desired outcomes and content of the interactive, computer-based lesson, for ensuring application of learning and offering an additional opportunity for ongoing assessment, where desired.

The worksheet and task answers

Full answers for each worksheet and task are provided, where a definite answer can be expected. Where appropriate, there is also some explanation of how these answers are arrived at, and some indication of common misconceptions and errors. Where the task requires a more open-ended response, some sense of the sort of answer that might be expected is given.

Making the most of Mindset Learn IT video lessons

Videos are a wonderful resource for teaching and learning. They:

- bring the world to the classroom, through interviews, problem solving in real contexts and scenes that show places and activities beyond the classroom
- show learners how things work, and what they actually look like
- offer clear explanations, supported by good diagrams, animations and other visual supports

The Mindset Learn videos engage the learners actively by:

- posing questions for the viewer to think about while watching the video
- asking the viewers to anticipate what might happen, or to try something out for themselves, and to check their ideas and findings with what they see later in the video
- showing a variety of ideas and information, and asking learners to weigh the evidence and reach their own conclusions

These characteristics of the Mindset Learn video lessons help learners to understand what they are learning, and to see how it has relevance in the 'real' world. Thus, they motivate learners and help to make learning meaningful for them. In order to get the full benefit from viewing a video, however, learners need to incorporate the ideas developed in the videos into knowledge they already have, refining and extending their understanding. This means that they must engage actively with the material in the video. How well this happens is partly determined by the nature of

the video itself, but more, importantly, by the way teachers use the video.

To use videos well, teachers need to consider how to integrate the videos both into the learning programme as a whole, and into individual lessons.

Here are some useful questions to ask when planning how to integrate videos into a learning programme:

- Will we use the series of video lessons as the organising structure for the learning programme, or follow a different structure that still includes the video lessons?
- Will we use all the videos, or only those that meet specific needs at certain points along the way?
- Are there some sections of certain videos that are particularly useful? Can we use these and not the entire video?
- Could we use some of the methodology illustrated in a video without actually showing that video to the learners?
- What practical arrangements do we need to make in planning which videos to use, and where in the programme to use them (such as changes to the timetable and booking the venue)?

These questions might be useful in planning how to use a particular video in a lesson:

How will we introduce the video?

It needs to be linked into the work that has gone before, and learners need to understand where the content of the video fits into their present section of work. The teacher also needs to motivate learners to watch it actively and purposefully. Teachers might pose a question that the video helps explore, or ask the learners to watch out for something in particular.

What will learners do while watching the video?

Teachers need to control the viewing, and to plan where they might stop the video for some class discussion, to explain something, to check learners' understanding or to give learners a chance to try out something that is demonstrated in the video. There are many opportunities for these sorts of activities in each video. Most videos mark some of these places with a pause button – but teachers need to make their own choices depending on the needs of their learners.

What will learners do after watching the video?

Each video ends with a task and teachers can ask learners to complete them or choose to do other follow up work that will consolidate or extend the learning from the video.

The suggestions made here and in each of the teacher guides for video lessons, offer teachers help with their planning, both for a learning programme and for individual lessons. However, nothing will be as good as watching the videos, and sharing ideas with colleagues about which videos to use and how best to make use of them.

Introduction

Making the most of Mindset Learn IT interactive, computer-based lessons

The interactive PC lessons allow learners to work through sections of the curriculum, concepts and learning at their own pace. The medium is engaging and interactive, requiring learners to control the amount of time spent on each learning step, and providing opportunities for self-checking and remedial feedback.

The interactive, computer-based lessons are an excellent resource for teaching and learning. They:

- explain concepts and ideas in a step by step manner, allowing the learner to direct the pace of learning
- provide models or case studies in real life applications, while simultaneously stimulating learners to think of interesting approaches for the PATs
- Where possible, demonstrated to learners how things work, and what they actually look like
- offer clear explanations, supported by good diagrams, animations and other visual supports
- provide the opportunity for learners to go back to sections requiring more intensive work and/or revision

The Mindset Learn interactive PC lessons engage the learners actively by:

- showing learners explicitly what they will be learning in each lesson and indicating the lesson's case study or context in which the specific learning will take place
- locating the learning in unique, school-related or workplace- type learning contexts, making them interesting and applicable to a number of contexts
- providing regular, short 'quizzes' to check ideas and understanding, before proceeding to further learning
- ending each lesson with a structured check of the extent of the learner's learning and understanding, and an opportunity to return to sections requiring more attention

Educator should consider using the principles and questions described above for introducing video materials, for these interactive media. Learners could work through each lesson on their own or could work at a guided pace so that the class takes each step together, allowing for discussion and questions between teacher and learners. These interactive lessons also work very well as revision tools for all learners, and particularly for those experiencing a challenge with a specific concept.

The Mindset approach to Information Technology

The focus of the National Curriculum Statement (NCS) for IT is on activities that deal with finding solutions to problems through logical thinking, information management and communication. It also places emphasis on the development of computer applications using current development tools. The subject develops awareness and an understanding of the social, economic and other implications of using computers.

The Mindset Learn IT video series meet the NCS challenge by taking learners to many interesting places and giving them access to a wide range of people who can link classroom knowledge with the real-world. The video series addresses all the Assessment Standards of the Grade 10 outcomes for IT in the NCS.

This print module supports the IT video series. It provides information about the curriculum, about

the video lessons themselves, and offers teachers exciting ideas on how to teach new concepts and to help learner's master skills in an engaging manner.

To be successful in the IT subject it is necessary for learners to have a thorough understanding of all the learning outcomes:

Learning Outcome 1: Hardware and System Software
Learning Outcome 2: E-communication
Learning Outcome 3: Social and Ethical Issues
Learning Outcome 4: Programming and Software Development

Our IT materials address these in the following ways:

Learning Outcome 1: Hardware and System Software
There is one series on Hardware and Software related to this Learning Outcome, supported by video lessons. The Troubleshooting Hardware and System Software Problems builds on the hardware-related resources provided in the Mindset Learn Grade 10 and Grade 11 IT materials. In this Grade 12 series, the focus is on determining causes of hardware and system software problems, through collecting information, asking questions and following a step by step process of elimination. Possible causes are discussed and solutions are discussed and shown. The problem-solving covers main hardware components inside computers, as well as devices and peripherals. It also deals with problems encountered with software, networks and the Internet. Where possible, solutions are demonstrated in action or in the form of graphics.

Learning Outcome 2: E-communication

The short series on the Social and Ethical Impact of Information and Communication Technologies covers some assessment standards here: analysing information obtained from the Internet, determining one's internet security and the consequences of poor security measures, protecting the value and distribution of information produced by a person or company, and understanding how the latest technologies facilitate human interaction

Learning Outcome 3: Social and Ethical Issues
The short series about the Social and Ethical Impact of Information and Communication Technologies deals with the manner in which our lives have changed and continue to be changed by ICTs and offers ideas about some of the positive and negative consequences of ICTs in our lives, in society and to the environment, as well as how to use ICTs responsibly. There is also a lesson about threats inherent in the use of ICTs, such as privacy issues, cyber bullying, and malicious attacks in the form of viruses and related threats. The lessons provide ample opportunity for debate, and lend themselves to discussions about current trends and issues pertinent to learners' personal and work-related use of ICTs.

Learning Outcome 4: Programming and Software Development
There are three series pertinent to this learning outcome. The Database Applications series systematically guide learners through all aspects of database development and use. The Programming in Delhi and Java series covers many portions of programming, in user-friendly and well-paced lessons, with the option to choose the programming language in use in the school. Each of these programming lessons is based on a case study, frequently relating to real life programming needs in a school or work situation. The scenarios model the sorts of applications

of programming that learners may encounter, and might provide useful ideas for selecting

specific topics for the practical assessment task. There is also an extensive series covering the phases of the Practical Assessment Task. The lessons are intended to provide guidelines to enable learners to complete each step of the PAT in as much detail as possible, and to apply their database and programming knowledge thoroughly and effectively in the practical assessment task. As indicated, the first two phases are presented in video lessons, while phase 3 is in the form of interactive, computer-based lessons.

Using the Mindset IT Materials

The curriculum poses many exciting challenges for both IT teachers and learners, and it is therefore important to continually expand on knowledge and skills. Encourage your learners to meet these challenges by doing research and critically engaging in discussions on current trends and new developments.

It is important for you to plan how best to use these materials. You could choose to show your learners a whole series of lessons, or you could select individual lessons to address a specific topic or to demonstrate something that you could not replicate in the classroom. In selecting a series to show your learners you will notice that whilst each series can stand alone, there are logical connections between some series of lessons. For example, you will notice that the series on Database Applications links with the series on Programming, while the series on the PAT refers to concepts covered in the lessons in the Databases and Programming series.

You should also encourage your learners to try out some of the many demonstrations in the videos or computer-based lessons. For example in the Troubleshooting series, learners are shown how to fix certain problems. Use this opportunity for learners to apply what they have learnt, by replacing a component or changing settings as demonstrated in the videos. A useful exercise might be to try to apply the remedies shown, even if the school's hardware or software is different to that shown in the lessons.

Database Applications

Free your brain

Series overview

There are 6 lessons in this series and their purpose is to give learners an in-depth understanding of data storage and the construction, manipulation and maintenance of high quality relational databases. The series builds on the Grade 10 series, where learners were introduced to databases and shown how to construct single-table databases, and how to create forms, queries and reports. The Grade 12 series will give learners a clear idea of how to structure the data in a database so that these tools can be used effectively, to produce high quality information, as well as what steps must be followed to ensure that the structure is maintained and protected.

Curriculum links

These interactive, computer-based lessons achieve the outcomes of the NCS, and are also designed to cover the topics listed in the CAPS document, including:

- Describe database management software
- Concepts of database management - size and accessibility
- Data collection
- Import and Export data
- Create, design and modify a relational database
- Characteristics of a good database – data integrity, data independence, data redundancy, data security, ease of data maintenance
- Characteristics of quality data
- Data access and manipulation
- Engage with basic structured query language (SQL) code and manipulate a relational database
- Design and create queries - structured query language (SQL), using a join on a maximum of two tables with multiple criteria, using a simple WHERE clause
- Grouping data – records, fields, related data
- Data maintenance tasks
- Problems with databases – anomalies, removing anomalies through normalization, splitting tables and creating relationships, where does un-normalised data come from?
- Database design - primary and foreign keys, relationships, 1:M relationships, joins
- Simple entity relations diagrams (ERD)
- Protecting data – validation, verification, integrity, controlling access
- People and database management and development

Educational approach

Database storage is placed in context by showing how it relates to other data storage in computers, and showing why some data should be stored in database tables. Learners are shown how to design a well-structured database, and how to re-structure an existing, poorly designed database. The theory is supported by animations, showing learners how to use database management software tools to achieve their design goals and showing other practical examples of the content. The same set of data is referred to frequently, so that learners can focus on the concepts and tools, rather than the data itself. It is important that learners apply what they learn to new scenarios and different data sets, and educators are encouraged to create examples of tables with un-normalised or abnormal data for learners to re-structure.

Database Applications

Free your brain

Series at a glance

Lesson title	Lesson Outcomes By the end of this lesson the learner should be able to:
1. File Types and Data Storage	• Differentiate between different data types and identify the type of file which can store data of each data type • Explain why text files are used to exchange data between programs • Differentiate between 'good' and 'bad' data storage in terms of file storage on disc (permanent) and the use of RAM (temporary)
2. Database Tables	• Compare text files, spreadsheets, word processing tables and database tables in terms of how the stored data can be manipulated and queried • Critically evaluate a database table to identify data redundancy and data anomalies • Compile a list of the facts which will need to be stored in a database for a given scenario
3. Entity Relationship Modelling	• Identify and illustrate a top-down approach • Explain the terminology and symbols associated with ERM • Create an ERM diagram for a given scenario • Use an ERM diagram to construct tables, create relationships and enforce referential integrity
4. Normalising a Database	• Understand when and why normalisation is needed • Explain the terminology used in normalisation • Normalise the design of a database • Implement normalisation by splitting tables
5. Database Queries	• Understand the software components of a database query system • Locate and apply a GUI tool for queries • Distinguish between creating and running a query • Understand how relationships and joins affect queries • Select join types and create and run a query
6. Database Maintenance	• Understand the importance of taking steps to make it easy to work with data in tables, but at the same time ensure that data is secure • Create lookups to access data in related tables and simplify entry of data • Open a database for exclusive use, share data amongst forms as part of the solution to the problem of securing data, create a simple switchboard, and customise the navigation pane • Split a database into front-end and back-end databases, set a database password, and make a back up of a database

Database Applications

Teaching Guidelines

Lesson 1: File Types and Data Storage

This lesson is about the files created by different programs. All data is stored digitally (0s and 1s), and depending on the file type is interpreted in different ways by different programs. Metadata is stored by most programs, but not by text editors. This means that text files are portable, and can be used to transfer data between programs. Hard drive capacity is not a problem, but the limited amount of RAM means that data files should be kept as small as possible. The use of virtual memory to manage large data files will slow down processing.

Task

Learners must create and save files containing the same data, using several different programs. They must then investigate the **file size**, which indicates how much **secondary storage** or hard drive space is needed, and the amount of **primary memory** and **virtual memory** needed to open the related programs.

Notes and Teaching suggestions

The task is intended to give learners practical experience in examining data file sizes and RAM usage. We suggest that you let learners work in pairs to draw up the table, and use an informal assessment. Let one or two learners give their findings, and ask questions such as:

- Which program/s created the biggest files? (database)
- Why is it better not to have to use virtual memory? (It slows down the processing).
- Does the program or the data file use the virtual memory? (the program)

As another exercise, you could tell learners to open a large PowerPoint file and note the changes in RAM and virtual memory usagewhen they run a slide show. More virtual memory is used for running the slide show than for working on the presentation. This is because more processes (program instructions) must be used for a slide show. Point out to learners that the processes are the running programs – with .exe extensions.

You can also tell learners to open multiple programs at the same time to see the effect on the virtual memory. Note that the amount of virtual memory refers to hard drive space. If you keeping opening programs (trying to fill virtual memory), the operating system should eventually show a message that it is increasing the amount available as there is not enough. The operating system merely allocates more hard drive space for pages (blocks or units of program instructions from one program).

Related topics to teach before or after this lesson are virtual memory, file allocation tables and disk fragmentation (which slows down access to files stored on the hard disk).

Lesson 2: Database Tables

Here we compare data stored in a structured way by four different programs: Notepad, MS Word, MS Excel and MS Access. We determine that data to be used by DBMS software and SQL must be logically structured so that the tools can be used effectively. A poorly constructed table is used to point out redundancy and anomalies, and the nature of quality data is introduced. We show learners how to protect data from human error, and suggest an easy method for deciding which data to store in a given scenario.

Task

For the first task, learners must draw up a table of comparison for structured data storage. The second task requires that they inspect poor quality data to identify inaccurate, incomplete and inconsistent data, as well as incorrectly stored and redundant data.

Notes and Teaching suggestions
The first task should take only a few minutes to complete. We suggest that you let them do this as an individual task and electronically, creating a table in a document. They can illustrate the information by adding screenshots of each type of file, which will be easier done if they first complete the following related task.

Extra Task: Importing and exporting data.
Start by creating a spreadsheet in MS Excel, and saving it as a spreadsheet and then as a CSV file. Open the CSV file in Notepad, and save it as a text file. Create a new database in MS Access and import the CSV file, saving it as a table. Right click on the CSV file, select Open With and open the file in MS Word. It will open as text. Select the text, then on the Insert tab, select Table, then select Convert text to Table. Save the file as an MS Word document.
This exercise will give learners a clearer understanding of data storage and how to transfer data between programs.

The second task is intended to help learners identify poor quality data. If your learners have had little or no experience of analysing tables, it might be better to do this as a class exercise and show the given table or a similar one on the blackboard, OHP or using a data projector if you have one.

When looking at a field that stores more than one data item (job titles) make sure learners realize that you cannot access parts of a field. A field should represent only one attribute of an entity. Encourage learners to look critically at all database tables, those they are given and those they create.

Lesson 3: Entity Relationship Modelling
Learners are shown how to design a database from the top down, which should always be done before any data is captured. The purpose of the lesson is to help learners to decide which of the data in a scenario should and which should not be stored, and to determine how that data must be organised and stored, in order to achieve the goals of efficient data capture, ongoing accuracy and efficient querying and reporting. Entity-relationship terminology is explained, and they are shown how to construct an Entity Relationship Diagram (ERD).

Task
The learners must draw up an ERD and use the diagram to construct tables and establish relationships. The purpose of the task is to give them hands-on experience in applying the theory.

Notes and Teaching suggestions
This task will take more than one lesson to complete, and we suggest that you assess their diagrams before they start constructing the tables. Depending on the ability of your learners, it would also be a good idea to revise the setting of field properties before they create the tables. The most important are: input masks (to assist the user to use the correct format), validation rules (to ensure the input is in an acceptable range of values), required (to ensure the field cannot be left blank), caption (to show a more user-friendly field name or alias, as field names cannot contain spaces), and limiting the field size (to keep storage to a minimum).

It may help to remind learners that the foreign key is always in the table on the many side of the relationship, or that the foreign key always looks up one primary key in another table.
If you feel that the creating of the relationship by using the Lookup Wizard might be difficult for your learners, this step can be demonstrated first, or done by letting the learners as a class follow oral instructions from you as they do the work, once they have created all their tables.
Note: The creation of a Lookup table is demonstrated in the next lesson.

The relationship can also be created by dragging and dropping in the Relationships view, but this does not create a Lookup table.

Lesson 4: Normalising a Database

In this lesson, we show learners how to re-structure a poorly designed database table, for data that has already been captured. We explain how to identify repeating columns and rows, and dependent data. The three normal forms are explained, and animations are used to show them a structured and practical way to implement the splitting of the tables.

Task

Learners are given a table that must be normalised into third normal form (3NF). They are required to place all resulting tables in 3NF.

Notes and Teaching suggestions

The reason that we suggest that a spreadsheet be used to split the tables, is that it is far easier to manipulate the data, and that the data can be seen at all times. It is easy to check and correct things with a spreadsheet.

The Table Analyzer (and splitter) can only be used by someone who has a very clear understanding of normalised table structures. It really does not work well on a table with many anomalies.

If the tables are split manually in MS Access, one only sees messages like “You won’t be able to undo this” and “some of the data may be lost”. This is not helpful to learners.
We suggest that you have an initial class discussion about the table, bringing in the following:

- Which field names are the same or very similar? (Subject1 etc – this indicates repeating columns.)
- Is there a primary key? (No)
- Ignoring the Subjects columns, which data is repeated in the rows? (Grade, PostOffice, PostalCode.)
- Which of these belongs to the Learner entity? (Grade)
- Which data depends on one of the other fields? (PostOffice and PostalCode depend on each other – a new table is needed, Age depends on DOB - date of birth.)
- Are there values that can be calculated? (Yes. Age – so the field can be deleted)
- Does the PO Box number belong to the Learner or the Post Office? (The learner – so it should stay in the Learners table.)

If time is restricted, learners can do the exercise on paper. With sufficient time, we suggest that you supply the table (as shown to learners) in a database. Let them export the table to MS Excel and import the resulting tables to MS Access.

Lesson 5: Database Queries

In this lesson we show learners how DBMS and programming software manage queries. The role of the database engine and the query builder are explained. We attempt to answer the question, “What is a query?” so that learners can fully understand what they are trying to accomplish when they build and run a query. After explaining the difference between select and action queries, we show learners how to create and run queries on a single table in both MS Access and SQL. We then move on to queries on multiple tables, and show the effects of the relationship and the join type on a multiple table query.

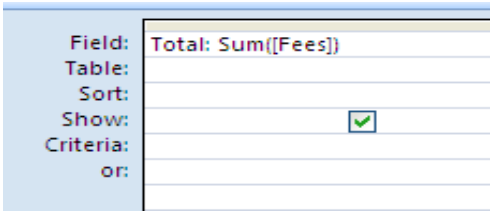
Task
Learners must draw up a one-page diagram to illustrate the different join types and summarise the effects of each one.

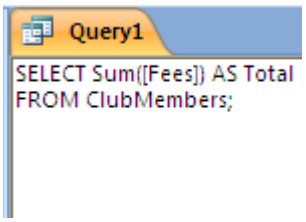
Notes and Teaching suggestions
In Database Applications learners usually experience the most difficulty with queries. We believe that this is often because they are shown how to do queries, without first ensuring that they know what queries are. In teaching the concept, we explain this, and show both the MS Access interface and the SQL statements that are used for queries.

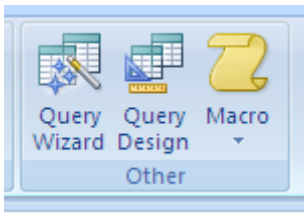
An action query modifies a table or the data it contains. A selection query returns a smaller *table* consisting of a copy of some of the data extracted from one or more of the tables (or from the result of another query). The data in a query can include a new field calculated from another field, or a query might contain a single new field, which summarises the values in another field. It is because selection queries have the same structure as tables (fields and records) that tables and queries are often dealt with in the same way. For example, you can create a query, form or report for a table or a query.

This is a complex lesson, and we suggest that you break it up and allow learners to do it over a few lessons. A possible sequence/division would be:

1. Database Engines and Query Builders (Activity: Complete the first question on the worksheet)
2. Select Queries, Action Queries and the related SQL commands -SELECT and INSERT, DELETE, UPDATE (Activity: Revise SQL clauses -give them a handout or list of SQL commands, clauses and formulae)
3. Creating and Running Queries (Activities: Complete the second part of the worksheet, then create a query of each type
   Note: The summary type of query can be created by selecting the Create tab and then Query Design. Add the table and type in the Field as below (centre). Alternatively, run a simple query choosing only the field to be summarised, and then change to SQL View to change the SQL statement, as in the last image. Run the query by changing to Datasheet View.







4. Relationships and Join Types (Activities: Learners do the task, and then using two simple tables like the ones in the lesson, let them practice queries using different join types)

Lesson 6: Database Maintenance
It must be as easy as possible to work with the data in a database. However, there will also have to be restrictions in the way that users can access some or all of the data, as well as putting measures like passwords and backups in place to secure the entire database, so that the data, the database structure and the database itself are protected.
In this lesson we show learners how to create Lookup tables to make data capture easier and to avoid errors. We explain why access needs to be restricted, and then show a large variety of ways in which this can be done. Once the database design has been implemented, we show them how to set passwords, do backups and split databases.

Task

Learners must apply as many of the techniques from the lesson as possible to an existing database. They do not have to do passwords, backups or database splitting. The purpose of the task is to make learners investigate all the techniques in a real-world situation. The application of a technique should consolidate the learning, and encourage learners to see the reasons for using that technique.

Notes and Teaching suggestions

Lessons 2, 3 and 4 focus on tables, the core of the database, and the structures for storing all the data. Lesson 5 deals with queries. This lesson explains what forms and reports are and why we use them. It is important that learners' understanding of the two goes beyond the following common misconceptions:

- A form is just a more user-friendly way to capture data.
- A report is used to prepare output for printing.

The lesson explains how forms and reports, as well as other tools and settings are used to ensure that we:

- maintain the database structure
- protect sensitive data
- prevent accidental or malicious deletion of data

This lesson can be separated into three logical sections:

1. Using Lookup tables to create the relationship between the FK field and the PK field for ease of data capture and to ensure data integrity; ensuring referential integrity by editing the relationship, and choosing whether to Cascade Update Related Fields or Cascade Delete Related Records; hiding fields in a table to restrict access in Datasheet View.
   Suggested activities: The first two parts of the worksheet, followed by a practical exercise, creating a lookup table and editing the relationship.
2. Restricting data access to parts of the database. This includes forms, reports, switchboards and the navigation pane, as well as Windows Options settings.
   Suggested activity: Learners draw up the list in the third part of the worksheet while they do the lesson.
3. Protecting the entire database against unwanted access and data loss: passwords, backups and splitting of a database into front-end and back-end.
   Suggested activity: A practical exercise implementing the ideas on an existing database.

For the task, we suggest that you first let learners self-assess their list of items. Before they tackle the task, hand out a checklist (next page) containing the items listed. Let learners do the task as a competition in pairs. They must then let another pair assess their efforts.

Database Applications

Learners doing the task tick the done column. Learners assessing the task give a mark (Example: 0 = not done, 1 = done but not very well, 2 = done well, 3 = an exceptional effort OR a very good new idea).

No	Done	Technique	Mark
1		Create a Lookup table	
2		Hide the primary key	
3		Hide columns in a table	
4		Enforce Referential Integrity	
5		Cascade Update Related Fields	
6		Cascade Delete Related Records	
7		Create a Form, omitting sensitive data	
8		Create a Report, omitting sensitive data.	
9		Create a switchboard	
10		Make the database open on the switchboard	
11		Hide the Navigation Pane	
12		Customise the Navigation Pane	
13		Change setting for: Enable design changes for tables	
14		Change the Application Title	
15		Change the Application Icon	
16		Create a Shortcut; store it in a different folder from the database.	
17		Change setting for: Hidden property of the database file	
18		Change setting for: Display Document Tabs	
19		Disable the MS Office Ribbon Tabs	
20		Customise the Quick Access tab to hold only essential tools not on Home	
Other			
Other			

Database Applications

Free your brain

LESSON 1 WORKSHEET
FILE TYPES & DATA STORAGE

List the five file types shown in the lesson and write down a possible file extension of each type.

1. _____
2. _____
3. _____
4. _____
5. _____

What is metadata?

_____

Explain why text files are used to transfer data from one program to another.

_____

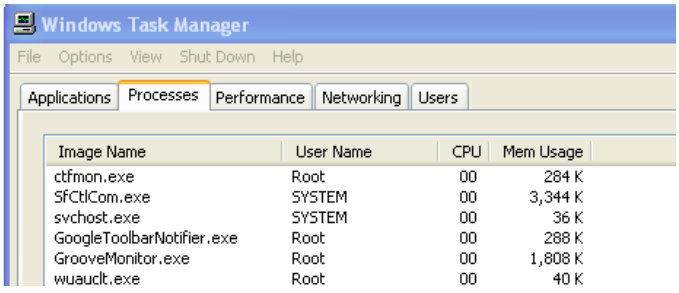
Lesson Outcomes

- Differentiate between different data types and identify the type of file which can store data of each data type
- Explain why text files are used to exchange data between programs
- Differentiate between 'good' and 'bad' data storage in terms of file storage on disc (permanent) and the use of

Task

Create four or five data files in different programs that consist of only your name and surname. For example, use Notepad, MS Word, MS Access, MS PowerPoint and MS Excel. Close the files and write down the actual size of each data file (Right click, Properties OR View, Details in Explorer).

Use Ctrl + Alt + Del to open the Task Manager. Click on the Processes tab, and then click on Mem Usage to sort the processes according to how much RAM is being used by each program. Open the files one at a time, and write down your findings, converting K (kilobytes) to megabytes. Close all programs, except the Task Manager. Click on the Performance tab and read the Page File Usage History, which shows how much virtual memory is being used. Open your files again, one at a time, and write down your findings about the use of virtual memory. (Hint: Subtract original amount)



Use your findings to draw up a table, with these column headings:

Program name **Virtual Memory Used** **RAM Used** **Data File Size**

Database Applications

LESSON 2 WORKSHEET
DATABASE TABLES

What is the purpose of a DBMS?

_____

_____

_____

List and very briefly explain three properties of high quality data.

1. _____

2. _____

3. _____

Lesson Outcomes

- Compare text files, spreadsheets, word processing tables and database tables in terms of how the stored data can be manipulated and queried
- Critically evaluate a database table to identify data redundancy and data anomalies
- Compile a list of the facts which will need to be stored in a database for a given scenario

Task

1. Complete the table below to summarise structured data storage:

File Type	text file	document table	spreadsheet	database table
Organised using	lines and separators			
Normally used by	Notepad			
Normally used for	simple data storage and portability			
Main data type	text			
Data manipulation tools provided by the interface?	none			
Can be queried using SQL?	no			
Can be used by Java or Delphi?	yes			

2. Make a list of the anomalies in the poor quality data in the table below:

ID	Employer	Employee1	Employee2	Employee3	Job Titles
1	ABC	John Moloi	Sarah		Driver/Sales lady
2	DEF	Jones Ben	Fulu Mswaki	Mr Selati	Salesperson x 2/Driver
3		Susan Fram	PPercy Maki	Moses Dibe	Driver/Driver/Salesperson
4	JKL	Roy	Frank	Mohamed	Salespeople

Database Applications

LESSON 3 WORKSHEET

ENTITY RELATIONSHIP MODELLING

Lesson Outcomes

- Identify and illustrate a top-down approach
- Explain the terminology and symbols associated with ERM
- Create an ERM diagram for a given scenario
- Use an ERM diagram to construct tables, create relationships and enforce referential integrity

Briefly explain how each of the following is achieved:

Efficient data capture

_____

_____

Ongoing accuracy

_____

_____

Efficient querying and reporting

_____

Place the following steps in the correct order for a top-down database design:

Add foreign keys and create Lookup tables	1. _____
Enforce referential integrity	2. _____
Create relationships	3. _____
Identify the tables required	4. _____
Identify relationships	5. _____
Identify the fields required	6. _____
Add primary keys	7. _____
Set properties to maintain the field structure	8. _____

Task

A database is needed to store data about soccer players and the clubs they belong to. Annual membership fees vary from R150.00 to R350.00. Using the list of fields given below, do the following:

1. Draw an Entity Relationship Diagram showing the entities, their primary, natural and (if needed) foreign attributes, and the relationship between the entities including the one and many sides of the relationship.
2. Create the two tables in a database. Choose appropriate data types and set field properties such as field sizes, input masks, and validation rules to ensure efficient data capture. Set the primary key for each table. For the foreign field, use the Lookup Wizard to allow the field to look up the primary key in the other table. Close both tables and use the Relationships view to edit the relationship and enforce referential integrity.

FirstName	AnnualFee	ClubName	DateOfBirth	ClubID
Location	PlayerID	PlayerCellNo	Manager	ManagerCellNo

Database Applications

Free your brain

LESSON 4 WORKSHEET

NORMALISING DATABASES

Fill in the missing words to describe the three situations that show that data is "abnormal" for a DBMS and must be normalised:

1. There are _____ and _____ in a table, and queries do not work properly.
2. There is a _____ to _____ relationship between two tables.
3. A _____ has attributes which must be stored.

Lesson Outcomes

- Understand when and why normalisation is needed
- Explain the terminology used in normalisation
- Normalise the design of a database
- Implement normalisation by

Which view, Design or Datasheet, is most helpful to inspect a table for each normal form?

1NF	move repeating columns to new table/s, assign PKs to all tables	_____
2NF	move columns with repeating rows to new table/s, assign PKs to new table and FKs to original table	_____
3NF	move columns not dependent on the PK to a new table, assign PKs and FKs	_____

Task

Smart Students, a non-profit organisation, aims to assist learners who want to do well at school. For a small fee, learners will have Grade 10, 11 or 12 study notes for their three chosen subjects mailed to them. Normalise the table below. The resulting tables must all be in 3NF, Datasheet View, Design View

Note: A learner can choose a combination of three subjects, and more combinations than those shown will be offered. For example: Combination 1 is Geography, History and IT and Combination 2 is Geography, History and Physical Science.

LearnerName	Grade	DOB	Age	POBoxNo	PostOffice	PostalCod	Subject1	Subject2	Subject3
Mpho Radebe	10	1996/09/20	15	245	Horizon	0001	Geography	History	IT
Jonas Mashele	11	1995/04/28	15	168	Dale Park	0002	Phys Scienc	Geography	History
Mary Isaacs	12	1994/05/17	17	34	Dale Park	0002	Geography	History	IT
Robert Jones	11	1994/12/14	16	643	Horizon	0001	Geography	History	IT
Sheila Tovey	10	1996/11/03	15	381	Sun View	0003			
Henry Dibe	10	1996/02/07	15	67	Sun View	0003			

Field Name	Data Type
LearnerName	Text
Grade	Text
DOB	Date/Time
Age	Number
POBoxNo	Text
PostOffice	Text
PostalCode	Text
Subject1	Text
Subject2	Text
Subject3	Text

Storing numbers

If you do NOT need to do calculations, always use the **text** data type for numbers eg. PO Box Number and Postal Code.

Database Applications

LESSON 5 WORKSHEET
DATABASE QUERIES

Lesson Outcomes

- Understand the software components of a database query system
- Locate and apply a GUI tool for queries
- Distinguish between creating and running a query
- Understand how relationships and joins affect queries
- Select join types and create and run a query

Briefly describe the purpose of the following:

A Database Engine _____

_____

A Query Builder _____

_____

Remember: A select query selects some of the data and displays it as information, but does not create, modify or delete fields or records in tables. An action query alters the structure of the fields or records in tables.

Write one of the phrases from under the table in each space to show the 3 select query types:

Purpose of the Query	SQL
	SELECT Name, Surname, DOB FROM Learners WHERE YEAR(DOB) > 1990
	SELECT Name,Surname,MONTH(DOB) AS [Birthday Month] FROM Learners
	SELECT FORMAT(SUM(Fees),"Currency") AS [Total] FROM ClubMembers

- summarise values in one field
- show specific fields or specific records
- calculate new values

Task
Draw up a one-page summary of the Join Types in any visual format.
For example, you can use a table, or divide your page into 4 sections, or use a mind map with the heading, Join Types, at the centre.
Include the following in your summary:

- The name of each join type
- A set diagram for each join type
- Whether there is a relationship, and if so, which is the one and which is the many side.
- Labels: Primary Table/Left Table, Related Table/Right Table, PK fields, FK fields
- A sentence describing which fields, if any, are included in the join (intersection) of the sets.

Database Applications

LESSON 6 WORKSHEET
DATABASE MAINTENANCE

It must be as easy as possible to work with the data, but there will also have to be some restrictions in the way that users can access some or all of the data. Give three reasons why you should restrict access to data.

1. _____
2. _____
3. _____
_____

What is the main advantage of using Lookup tables for data capture?

_____

_____

List as many ways as possible of protecting the data in a database. (Your three reasons will help to identify these.)

_____ _____
_____ _____
_____ _____
_____ _____
_____ _____
_____ _____
_____ _____

Lesson Outcomes

- Understand the importance of taking steps to make it easy to work with data in tables, but at the same time ensure that data is secure
- Create lookups to access data in related tables and simplify entry of data
- Open a database for exclusive use, share data amongst forms as part of the solution to the problem of securing data, create a simple switchboard, and customise the navigation pane
- Split a database into front-end and back-end databases, set a database password, and make a back up of a database

Task
Working in pairs, use the list you created or one supplied by your teacher, and any well-constructed multi-table database. Apply as many of the techniques on your list as possible to your database. Your goal is to make your database easy to use AND at the same time, to protect the structure and contents of your database.

When you have finished the task, another pair of learners must test your database to see if it is really easy use and "bullet-proof". You will also be testing someone else's database, and in this case, you must deliberately (or maliciously) try to get access to "forbidden" data, and to delete or modify data. To keep the competition fair, you may not change or undo any of your "rival's" attempts to protect their database.

Database Applications

LESSON 1 WORKSHEET ANSWERS

List the five file types shown in the lesson and write down a possible file extension of each type.

1. text .txt
2. graphics .jpg
3. animation .swf
4. sound .mp3
5. video .avi

What is metadata?

Metadata is data about the data in the file (Example: instructions for how the data is to be interpreted, formatted or displayed), and data about the file properties (Example: file size, read only, date created)

Explain why text files are used to transfer data from one program to another.

Text files are used to transfer data between different programs (they make data portable), because they contain only data, with no metadata.

TASK ANSWERS

The amount of virtual memory used with none of the programs open was 139 MB.

Program name	Data File Size	RAM used by program	Virtual Memory used
Notepad	1 KB	3,6 MB	143 – 139 = 4 MB
MS Word	11 KB	28,3 MB	156 – 139 = 17 MB
MS Excel	9 KB	19,2 MB	154 – 139 = 15 MB
MS Access	324 KB	18,0 MB	152 – 139 = 13 MB
MS PowerPoint	33 KB	21,7 MB	153 – 139 = 14 MB

Notes:

The RAM and virtual memory amounts shown are the space used by only the program. These data files take up very little space in RAM (not shown in the Task Manager), but the database file is about 100 times bigger than the largest of the other files!

The much smaller amount of RAM and virtual memory used by Notepad, shows it is a much less complex program than the others. It has fewer processes that can be run. The more features (processing tools or instructions) there are in a program, the more space it will need.

LESSON 2 WORKSHEET ANSWERS

What is the purpose of a DBMS?

To control the creation, maintenance and manipulation of a database

List and very briefly explain three properties of high quality data.

1. Accurate – the facts are correct
2. Complete – we have all the data

3. Consistent – the data is stored in the same way for different records. There is no redundancy (each fact is stored in only one place) and all data is only dependent on the primary key.

Database Applications

TASK ANSWERS

1.

File Type	text file	document table	spreadsheet	database table
Organised using	lines and separators	rows and columns	rows and columns	records and fields
Normally used by	Notepad	MS Word	MS Excel	MS Access
Normally used for	simple data storage and portability	neat and readable presentation	calculations	data storage and querying
Main data type	text	text	numbers	any
Data manipulation tools provided by the interface?	none	limited	several	all possible tools
Can be queried using SQL?	no	no	no	yes
Can be used by Java or Delphi?	yes	no	no	yes

2.

ID	Employer	Employee1	Employee2	Employee3	Job Titles
1	ABC		Sarah ✔	✔	Driver ✔ /Sales lady ✔
2	DEF	Jones ✔	Fulu Mswaki	Mr ✔ Selati	Salesperson ✔ x 2/Driver ✔
3			P ✔ Percy Maki	Moses Dibe	Driver/Driver/Salesperson ✔
4	JKL	Roy ✔	Frank ✔	Mohamed ✔	Salespeople ✔

List of anomalies:

- The job titles are repeated and they are inconsistent (e.g. Sales lady, Salesperson, Salespeople.
- The job titles field contains multiple data items. It would not be possible to list the names of all drivers or all salespersons in a query.
- The data is incomplete, as one employer name and some of the surnames are missing.
- The names are inconsistent, as one surname (Jones) appears first, and one person has a title (Mr) instead of a first name. It would not be possible to sort the records by first name or surname.
- Percy Maki's name is incorrectly spelled as PPercy.

LESSON 3 WORKSHEET ANSWERS

Briefly explain how each of the following is achieved:

Efficient data capture is achieved by ensuring that each data item has to be entered only once, and the format is validated before the data is accepted.

Ongoing accuracy is achieved by ensuring that if data changes, it needs to be updated in only one place.

Efficient querying and reporting is achieved by ensuring that there is high data quality so that the information you need can be retrieved quickly and easily by the DBMS tools, in the format you require.

Place the following steps in the correct order for a top-down database design:

Add foreign keys and create Lookup tables 1. Identify the tables required

Database Applications

Enforce referential integrity
Create relationships
Identify the tables required
Identify relationships

Identify the fields required
Add primary keys

Set properties to maintain the field structure

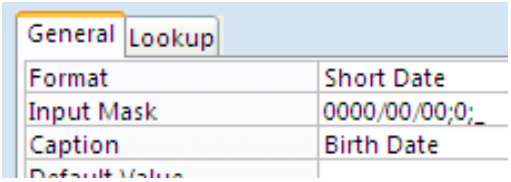
2. Identify the fields required
3. Set properties to maintain the fieldstructure
4. Add primary keys
5. Identify relationships
6. Add foreign keys and create Look up tables
7. Create relationships
8. Enforce referential integrity

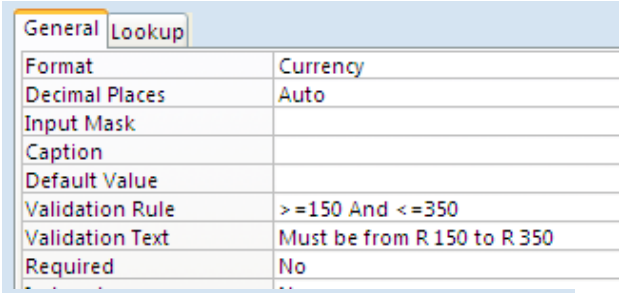
TASK ANSWERS

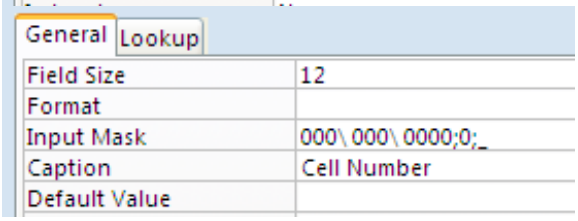
1. ERD for Players and Clubs

2. Database tables for Players and clubs

FirstName	AnnualFee	ClubName	DateOfBirth	ClubID
Location	PlayerID	PlayerCellNo	Manager	ManagerCellNo







LESSON 4 WORKSHEET

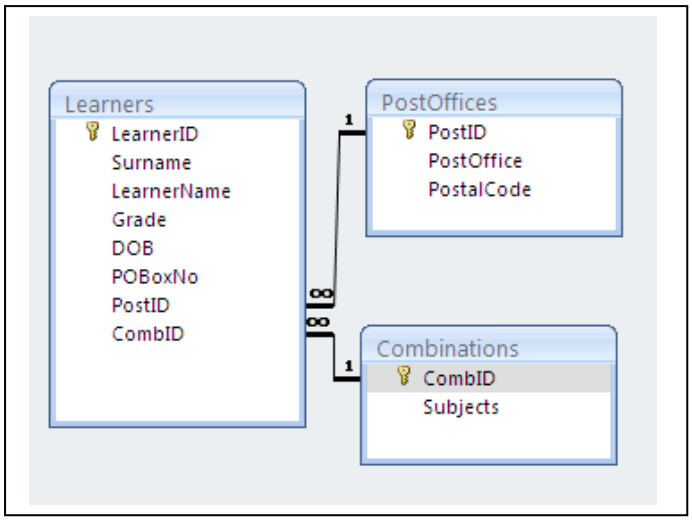
Data is “abnormal” for a DBMS and must be normalised when:

1. There are anomalies and data redundancy in a table, and queries do not work properly.
2. There is a many to many relationship between two tables.
3. A relationship has attributes which must be stored.

The most useful view to inspect a table for:

1NF is Design View — Repeating columns refers to repeating fields names.
2NF is Datasheet View — Repeating rows refers to the data values in the fields.
3NF is Datasheet View — It is easier to see if one field depends on another if you can see the data.

Database Applications



PostOffices

PostID	PostOffice	PostalCode
1	Horizon	0001
2	Dale Park	0002
3	Sun View	0003

TASK ANSWERS

Learners

LearnerID	Surname	LearnerName	Grade	DOB	POBoxNo	PostID	CombID
1	Radebe	Mpho	10	1996/09/20	245	Horizon	Geo/His/IT
2	Mashele	Jonas	11	1995/04/28	168	Dale Park	Geo/His/PhS
3	Isaacs	Mary	12	1994/05/17	34	Dale Park	Geo/His/IT
4	Jones	Robert	11	1994/12/14	643	Horizon	Geo/His/IT
5	Tovey	Sheila	10	1996/11/03	381	Sun View	Geo/His/PhS
6	Dibe	Henry	10	1996/02/07	67	Sun View	LSci/CAT/Ac

The postal code is stored as text. The subjects won't have to be accessed separately, so only one field is required. The way the address and subjects are stored is abnormal for humans, but it is normal for the DBMS. If a new post office or a new combination is added, it will be stored only in that table. This means there will be ongoing accuracy.

The name field has been split into two fields, as we may need to access the surname separately. In the Learners table, the user can look up the correct values for PostID and CombID from the other tables when the data is captured, which means there will be efficient data capture.

LESSON 5 WORKSHEET ANSWERS

A Database Engine manipulates the data in a database created by a particular database program.

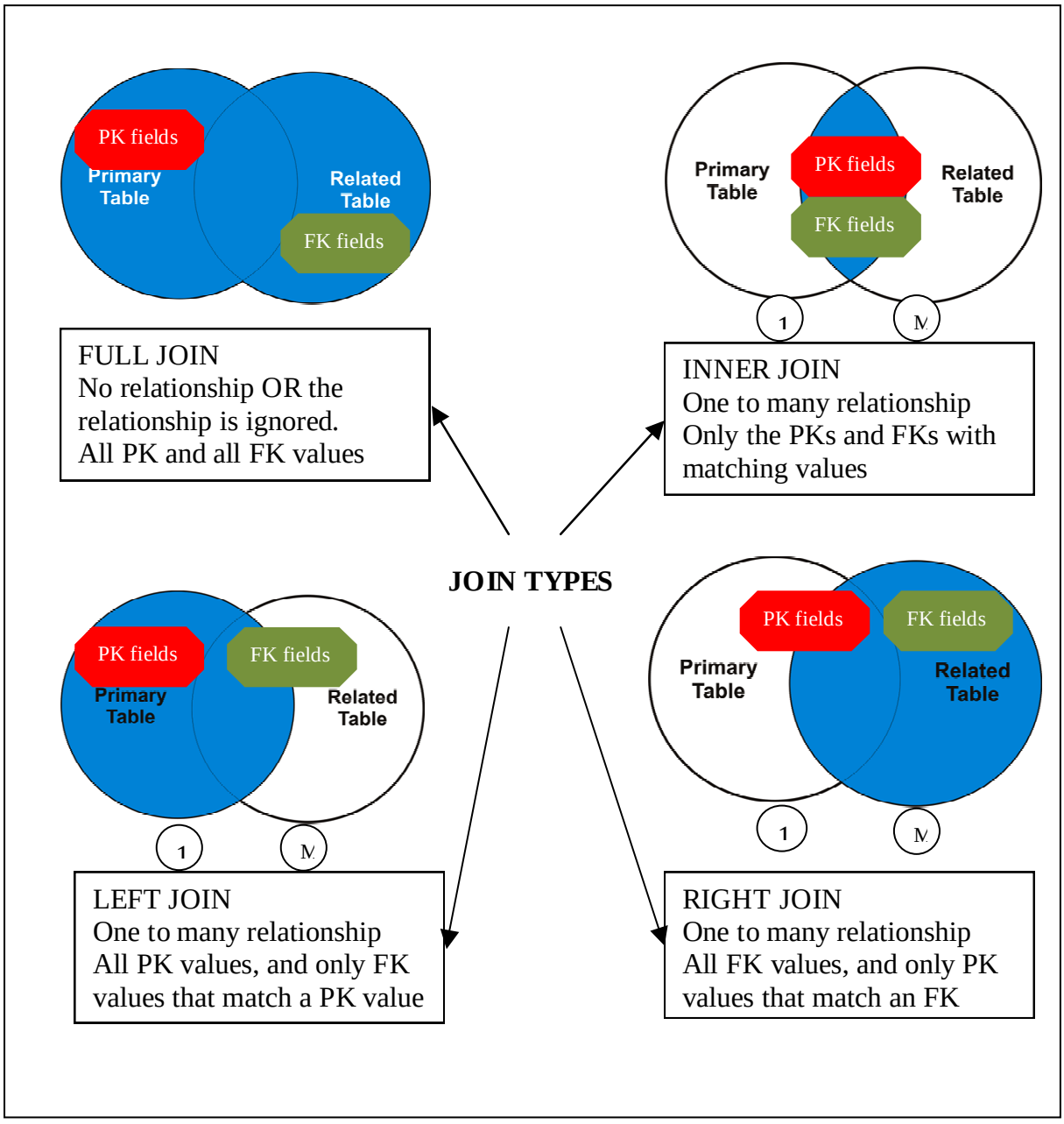
A Query Builder builds SQL commands into a string, and sends them to the database engine.

Select Query Types:

Purpose of the Query	SQL
show specific fields or specific records	SELECT Name, Surname, DOB FROM Learners WHERE YEAR(DOB) > 1990
calculate new values	SELECT Name, Surname, MONTH(DOB) AS [Birthday Month] FROM Learners
summarise values in one field	SELECT FORMAT(SUM(Fees),"Currency") AS [Total] FROM ClubMembers

TASK ANSWERS

See model answer:



Database Applications

LESSON 6 WORKSHEET ANSWERS

Data access should be restricted in order to:

1. maintain the database structure
2. protect sensitive data
3. prevent accidental or malicious deletion of data.

What is the main advantage of using Lookup tables for data capture?

Lookup tables make data capture easier, as the user can select the correct value from a drop-down list, instead of typing it in, which could lead to errors in the data.

List as many ways as possible of protecting the data in a database. (Your three reasons will help to identify these.)

- Create a Lookup table and hide the primary key.
- Hide columns in a table.
- Create a Form if the user may modify fields and records, or add records. (Delete is disabled by default.) Omit sensitive data
- Create a Report if the data must be read-only, omit sensitive data. The user cannot modify the structure or the data
- Create a switchboard so the user can only navigate to areas they may access
- Change settings to make the database open on the switchboard
- Hide the Navigation Pane
- Customise the Navigation Pane if the user needs to work with tables or queries
- Set Enable design changes for tables to false
- Change the Application Title and Icon so the user can't identify the actual database file
- Create a Shortcut to the database, store it in a different folder from that holding the database. Then right click on the database folder, choose Properties and set the Hidden property to true
- Set Display Document Tabs to false
- Disable the MS Office Ribbon Tabs, which leaves only Home and the Quick Access Menu. The Quick Access menu can be customised to hold any necessary tools that are not on the Home tab.

TASK ANSWERS

There is no model answer for this task. The checklist in the Teacher's Guide will give an idea of what may be included.

The database should be easy to use, and should have as many of the techniques / features as are possible. Remember that this was just an exercise. In real life, you should not add a technique simply for the sake of using it. Consider who the end-user of the database will be, and add techniques or adjust settings accordingly. Make sure you know that some data must not be seen, but all data should be protected from accidental deletion or modification.

Programming – Delphi/Java

Series overview

There are 20 lessons in this series on Programming in Delphi and Java. Their purpose is to give learners a good understanding of all programming requirements and activities, be they in the Delphi or Java programming language. The Programming lessons in Mindset Learn's Grade 10 and 11 IT series may provide useful revision for some aspects of the Grade 12 Programming curriculum. A variety of core development tools gives learners the required ability to design and develop appropriate computer-based solutions to specific problems through programming.

Curriculum links

These interactive, computer-based lessons achieve the outcomes of the NCS, and many of the topics listed in the CAPS document. Mindset Learn's materials will be reviewed and aligned to the CAPS in time for its implementation.
This series deals with Learning Outcome 4: Programming and Software Development. It satisfies Assessment Standard - The learner is able to design, implement, test and deliver efficient and effective solutions to problem situations.

The learning outcomes of each lesson are listed in the section below called Series at a Glance.

Educational approach

All main concepts and activities relating to programming for the development of specific solutions are covered in discrete lessons. It is unlikely that the 20 lessons in this series will match the manner in which the teacher covers all programming requirements in the classroom. However, each lesson and topic works on a real-life solution development, whether in a school-like case study or something more allied with the working world. This makes them practical, interesting and applicable. It also means that teachers can select relevant portions of the lessons or allow learners to dip into them when they have specific questions and problems or provide opportunities for learners to use the individual lessons for revising and consolidating their understanding of programming.

There is a strong link between the Programming lessons and the series on the Practical Assessment Task, as well as with the Database Applications series. Most of the programming lessons are likely to be useful for stimulating ideas about learners PATs.

Worksheets can be used with these interactive, computer-based lessons, to more fully engage the learner and provide a means of recording some of the main ideas of each lesson. Each worksheet also offers a task at the end, to consolidate learning, as indicated in the Teaching Guidelines section of this guide.

Programming – Delphi/Java

Series at a glance

Lesson title	Lesson Outcomes By the end of this lesson the learner should be able to:
1. Simple Data Types	• Identify data types and sample data associated with them • Illustrate how to declare and assign values to variables using simple data types • Show the use of simple data types in programs
2. Control Structures: Selection	• Be comfortable using relational and Boolean operators • Be comfortable using control structures and use the if statement • Identify form of the switch/case statement and write code with the switch statement
3. Control Structures: Sequence	• Demonstrate logical flow of events in a sequential statement • Use a pseudo code algorithm to represent logical flow of events • Write sequential statements that initialise, input data, process, output, store and terminate
4. Control Structures: Repetition	• Write repetition code using the FOR loop • Write repetition code using the While Loop • Write repetition code using the Do, while/ Repeat…until Loop
5. Task Analysis I	• Identify ways of writing questions for input, process and output tables • Analyse a problem and come up with input, processing and output requirements • Convert program specifications from IPO tables to write a simple program
6. Task Analysis II	• Write trace tables and illustrate ability to track the current values in program fields • Use flow charts and recognise diagrams used by Input, processing and output • Design simple diagrams showing a decision making process involving simple selection and looping • Use UML and create UML diagrams to represent classes and objects • Analyse a problem and write pseudo code to assist in developing a program
7. Functions and Procedures I	• Apply the encapsulation principles in programming • Show how to pass parameters in methods and use correct return types • Write methods with encapsulation
8. Functions and Procedures II	• Understand the meaning and importance of modularity in programming • Understand the concept of variable scope in programming • Distinguish between value and reference parameters
9. String Manipulation	• Recognise the string classes and be able to construct strings

	• Write code using string methods for string manipulations and comparison • Manipulate strings by replacing characters, converting to upper and lower case, performing concatenation and removing spaces
10. Arrays I	• Declare an array, set its size and populate the array • Write code to search through an array • Create and use a constants array
11. Arrays II	• Use methods to calculate totals for array items, rows and columns • Write code to delete the array elements • Use methods to sort elements of an array
12. Arrays III	• Identify the basic concepts of a two dimensional array • Show ability to declare and populate a two dimensional array • Illustrate ability to iterate through a two dimensional array to list array elements
13. Classes and Objects I	• Define and instantiate a class and declare and initialise instance variables • Explain how to access data members and change data for fields • Compare class instances and objects
14. Classes and Objects II	• Define a class to represent an object • Create and dynamically populate an array of objects • Write code to search an array of objects
15. Classes and Objects III	• Give a detailed explanation of polymorphism and inheritance • Apply inheritance in classes • Apply polymorphism in classes
16. Text Files	• Identify basic key words for text files and file data access • Instantiate a text file and write data to a text w\file • Read multiple lines of text from a text file
17. Evaluating	• Explain what an exception is • Describe exception handling techniques • Use exceptions and handlers • Define documentation and use good documentation convention • Explain different kinds of testing, test cases and test plans • Explain bugs, debuggers and debugging
18. Graphic User Interface I	• Identify the Graphical User Interface (GUI) and design principles • Identify components in Delphi • Use Netbeans or Delphi and GUI components to develop a data input form

19. Graphic User Interface II	• Show an understanding of the event delegation model in Delphi • Recognize event classes in Delphi
20. SQL	• Explain what SQL is • Explain how Delphi or JAVA connects to and uses the database • Write SQL statements in Delphi or JAVA

Teaching Guidelines

Lesson 1: Simple Data Types
This lesson is about identifying data types and sample data associated with these data types. The lesson also deals with declaring and assigning values, and using simple data types in programs. The learning is centred around a practical example, in which Mr Khumalo starts with a class survey and goes on to show how information can be stored.

Task
Learners must analyse code provided, identify errors and provide correct code for these areas.

Lesson 2: Control Structures: Selection
This lesson deals with relational and Boolean operators, control structures, if statements and switch cases. The scenario used to convey this learning is that Chang asks Faizel to help him develop software to manage his uncle's shop better.

Task
For the task, learners must use nested if statements to write a program to display the given output, and must use a case statement in a specific practical way.

Lesson 3: Control Structures: Sequence
In this lesson, the main content is sequential statements, pseudocode algorithms for logical flow of events, and writing sequential statements to initialise, input data, process, output and terminate. Claire starts to help her mom in her restaurant, so that she can learn how to do things with computer software instead of manually.

Task
In the task, learners use sequential flowcharts/flow diagrams and pseudocode.

Lesson 4: Control Structures: Repetition
The content of this lesson is repetition – using the for loop, the while loop and the do...while or repeat...until loop. Learning is demonstrated through a scenario in which Thabo starts to write software for the school's coach, to help him to manage the 1st football team.

Task
The task gives learners 3 simple situations in which to write code for a part of a program that uses repetition.

Lesson 5: Task Analysis I
In lesson 5, the main learning points are about input, process and output – determining ways to come up with the input, process and output requirements, and then writing these for a program. Faizel goes to an IT company to job shadow, where he learns about how to develop a program to work out the invoice amounts for clients, and the commission for technicians who service the clients.

Task
Learners are given 3 different practical applications in which to practice drawing up IPO tables.

Lesson 6: Task Analysis II

More details of task analysis are presented in this lesson, as Fatima, who does the school's administration, asks Mr Khumalo to help her develop a program to do her job better and faster, and without so much paper work.

Task

For the task, learners must draw up a table of actors and use cases, as well as use a trace table to identify some coding faults.

Lesson 7: Functions and Procedures I

As Johan works with his uncle, Dr Rogers, designing a patient management system, learners see how to apply encapsulation principles in programming, pass parameters in methods, use correct return types and write methods with encapsulation.

Task

The task asks learners to identify various parts of a class diagram and code specific methods.

Lesson 8: Functions and Procedures II

The lesson is about the meaning and importance of modularity, variable scope and value and reference parameters in programming. Thabo has to manage all the sports teams in the school, because his football program worked so well. He continues to do more programming!

Task

Learners have to consider some code given, and predict the outcome.

Lesson 9: String Manipulation

In this lesson, learners get to deal with string classes, constructing strings, writing code for string manipulations and comparisons and manipulating strings. The scenario used to convey this learning is Claire visiting Tebogo at his place of work – and IT department in a media company.

Task

Create a flowchart that will carry out a specific function and have a specific outcome.

Lesson 10: Arrays I

Faizel writes a program for the Maths teacher. This sets the scene for learning about declaring an array, setting its size and populating the array. They also learn to write code to search through an array and create and use a constants array.

Task

Learners must write a loop to populate an array, display the integers in an array and complete a flowchart so that it searches in a particular way.

Lesson 11: Arrays II

This lesson is about using methods to calculate totals for array items, rows and columns, as well as writing code to delete the array elements and use methods to sort elements of an array. Faizel continues with this program for the Maths teacher, adding functionality.

Task

Learners must declare an array, write code to certain specifications and use a method to sort the elements of a specific array.

Lesson 12: Arrays III

This lesson deals with identifying the basic concepts of a two dimensional array, declaring and populating a two dimensional array, illustrating ability to iterate through a two dimensional array to list array elements. Claire continues with her software for her mom's restaurant, adding the functionality to control stock.

Task

For the task, learners must write the code to print a 2D array in rectangular form.

Lesson 13: Classes and Objects I

In this lesson, learners learn about defining and instantiating a class and declaring and initialising instance variables. They have to access data members and change data for fields, as well as compare class instances and objects. Tebogo spends the whole day with Johan at work, showing him how to use classes and objects.

Task

In the task, learners must write code for various methods to apply the lesson's learning.

Lesson 14: Classes and Objects II

The main focus of this lesson is to define a class to represent an object, create and dynamically populate an array of objects, and write code to search an array of objects. Thabo extends his sports management program to allow the coach to keep track of who wins the soccer league.

Task

The task gives learners practice in writing code to search through the array of objects to find a specific object.

Lesson 15: Classes and Objects III

In this lesson we deal with polymorphism and inheritance. Johan and Dr Rogers take this further, as they apply inheritance and polymorphism in classes.

Task

Learners have to explain polymorphism and fill in required terms.

Lesson 16: Text Files

The lesson covers the main aspects of text files: identifying key words for text files and data access, instantiating text files, writing data to text files, reading multiple lines of text from a text files. The scenario used to convey this content is Mrs Dube's school library, where Claire helps her to convert from a card system to a computerised book management system.

Task

For the task, learners must match columns of keywords for data access to text files.

Lesson 17: Evaluating

Chang and Faizel work through the program they developed for Chang's uncle, to sort out some of the bugs he has encountered. Learners will learn about exceptions, exception handlers, documentation convention, testing, test cases and test plans, as well as debuggers and debugging.

Task

The task asks learners to discuss the use of debuggers, how to set watches on software programs and how to add break points in the software program.

Lesson 18: The Graphic User Interface I

The lesson is about identifying GUI and design principles, as well as identifying specific requirements in Delphi and using Netbeans or Delphi to develop a data input form. Claire and Mrs Dube show us how it's done, for the new program they have designed.

Task

Learners have to evaluate a given graphic user interface.

Lesson 19: The Graphic User Interface II

In this lesson, learners find out about the event delegation model in Delphi and learn how to recognise event classes in Delphi.

Task

Learners apply their learning through matching columns and competing code.

Lesson 20: SQL

This lesson explains what SQL is, how Delphi and Java connect to and use the database, and gives learners the skills to write SQL statements in either Delphi or Java.

Task

Learners must use a given Access table to complete SQL statements for a variety of queries.

LESSON 1 WORKSHEET
SIMPLE DATA TYPES

To write a program to provide information about trees in the botanical garden, you need to declare some variables. Choose the most appropriate data type for the variables. Then choose variable names for these variables, declare the variables and assign the value *Acacia xanthophloea* to the name, 245 to the number of trees, 21.87 to the average height and NO or false to the Evergreen variable

Lesson Outcomes

- Identify data types and sample data associated with them
- Illustrate how to declare and assign values to variables using simple data types
- Show the use of simple data types in programs

Description	Data type	Declaration in Delphi	Declaration in JAVA	Assign the value in Delphi	Assign the value in JAVA
Name of the tree					
Number of this type of tree in the garden					
Average height for this type of tree in meters					
Evergreen or not					

Note: Name is a reserved word in Delphi and should not be used as a variable name

What is the biggest value that can be assigned to a variable of the type byte? _____

What happens if you assign a bigger value to the variable than what the data type allows?

_____

TASK

1. Find the errors in the following code:

DELPHI	*Corrected:*	JAVA	*Corrected:*

Var
x,y : integer;
z : integer;
begin
x : 5;
y = 6;
z := x/y;
lbloutput.caption := z;
end;
		Public static void main(string[] args) {
		int x;
		x = 5
		int y;
		y :=6;
		int z = x/y;
		system.out.println(z
		);	

LESSON 2 WORKSHEET

CONTROL STRUCTURES: SELECTION

Lesson Outcomes

- Be comfortable using relational and Boolean operators
- Be comfortable using control structures and use the if statement
- Identify form of the switch/case statement and write code with the switch statement

Fill in the appropriate relational operators for the expressions to be true:

Expression	Answers
8 10	
'Joe' ….'Jack'	
'A' 'a'	

Note: Letters and strings are ordered according to their ASCII values

What is the order of precedence when using boolean and relational operators? _____

_____

Determine the **outcome** (i.e. true or false) in the following cases if C = 10 and D = 20:

DELPHI	JAVA	
Outcome := (C<20) and (D<=20)	outcome = (C<20) && (D<=20)	
Outcome := NOT(C<D)	Outcome = !(C<D_	
Outcome := (C>=20) or (D<=20)	outcome = (C>=20) OR (D<=20)	
Outcome := (C>20) OR (D<300) AND NOT(C>D)	outcome = (C>20) OR (D<30)&&!(C<D)	

TASK

1. Use nested if statements to write a program to display the output "Distinction" if the mark is greater or equal to 80 and less than or equal to 100, "Pass" if the mark is between 40 and 79 (inclusive), "Fail" if the mark is between 0 and 39 (inclusive) and "Invalid" if the mark is > 100 or <0. Use 77 as input for the mark.

DELPHI	JAVA

2. Use a case statement or switch to display the season when the number of the month is entered:
   (Spring: 9,10,11; Summer: 12,1,2; Autumn: 3,4,5; Winter: 6,7,8)
   Use an integer value month with value 3 in your answer.

DELPHI	JAVA

LESSON 3 WORKSHEET

CONTROL STRUCTURES: SEQUENCE

Lesson Outcomes

- Demonstrate logical flow of events in a sequential statement
- Use a pseudo code algorithm to represent logical flow of events
- Write sequential statements that initialise, input data, process, output, store and terminate

Match the columns: Which symbols are used in flowcharts to indicate the following:

INPUT and OUTPUT		A 
DECISIONS		B 
PROCESSING		C 

What is the purpose of the arrows in flowcharts?

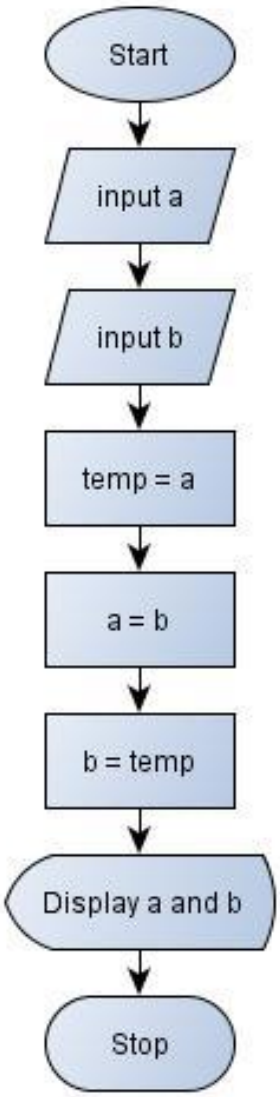
_____

TASK

1. Use a sequential flowchart to calculate the area of a rectangle.

2. Use pseudocode to plan the solution of finding the value of two values that were entered and display "equal" if they are the same.

3. Consider the flow diagram on the left and explain what the purpose of the process is:



_____

_____

LESSON 4 WORKSHEET

CONTROL STRUCTURES: REPETITION

Lesson Outcomes

- Write repetition code using the FOR loop
- Write repetition code using the While Loop
- Write repetition code using the Do..while/ Repeat…until Loop

Name 3 types of loops?

_____

Name the three parts of a for loop or the three things the for statement does.

_____

_____

Name the most appropriate loop:

Description	**Loop type**
To display “Welcome” 10 times	
To guess a number between 1 and 100 until you get it right	
To search though a list until you find a value or reach the end of the list	
To keep increasing a value by 10% if it is smaller than 50	

TASK

Complete the code for the following programs:

1. Code to display “Welcome” 10 times.

DELPHI	**JAVA**

2. Code to guess a number between 1 and 100 until you get it right.

DELPHI	**JAVA**

3. To keep increasing a value by 10% if it is smaller than 50.

DELPHI	**JAVA**

LESSON 5 WORKSHEET

TASK ANALYSIS I

Lesson Outcomes

- Identify ways of writing questions for input, process and output tables
- Analyse a problem and come up with input, processing and output requirements
- Convert program specifications from IPO tables to write a simple program

Categorise the questions that you can ask to determine the input, processing and output, when analysing a problem. Place the number of the question under the Input, Processing, Output or Storage column:

1. What is the data type of the input?
2. What kind of data does the input represent?
3. How can the answer be calculated?
4. What basic operators are needed?
5. What is the formula I need?
6. What is the data type of the output?
7. How is the answer to be displayed?
8. Do I have to print or display the data or store in permanently?
9. Do I need to format the output?
10. Where must the data be stored?
11. Do I have to print or display the data or store in permanently?
12. Do I need a loop?
13. Do I need if statements?

INPUT	PROCESSING	OUTPUT	STORAGE

TASK

1. **Complete IPO tables for the following:**
   a. Read the radius of a circle. Then determine and display the area of the circle.

I	P	O

   b. Read the length of three sides of a triangle. Determine if it is a right angled triangle and display a suitable message

I	P	O

c. Read a cell phone number. Determine if it starts with a 0 and is 10 characters long to determine if it is a valid cell number and display a suitable message.

I	P	O

2. Implement the IPO tables in the previous question:

LESSON 6 WORKSHEET
TASK ANALYSIS II

What is the purpose of the program (lower right), planned in the flowchart?

_____

What does UML stand for?

_____

Lesson Outcomes

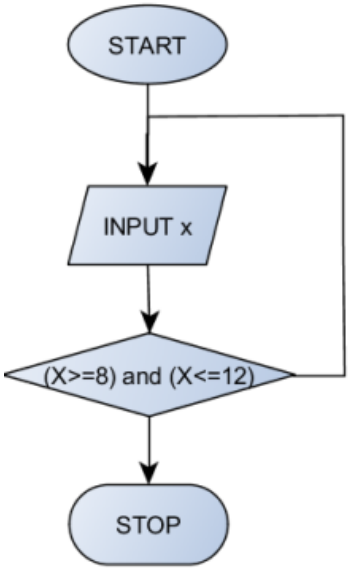
- Write trace tables and illustrate ability to track the current values in program fields
- Use flow charts and recognise diagrams used by Input, processing and output
- Design simple diagrams showing a decision making process involving simple selection and looping
- Use UML and create UML diagrams to represent classes and objects
- Analyse a problem and write pseudo code to assist in developing a program

TASK

1. Identify possible actors, communication and use cases in the following scenario:
   A client brings a pet to the vet. The vet examines the pet and prescribes medicine or performs a procedure if necessary. The client pays.

ACTORS	USE CASE

2. The following program (right) is supposed to input 10 values in a loop and find the smallest value entered. The output is incorrect. Use a trace table to show how the values of all variables in the program change as each line of code is executed and find the mistake. Explain what to do to debug the program.



Programming – Delphi/Java

DELPHI	JAVA

1. Var smallest, value, count : integer;
Begin
2. Value := strtoint(inputbox('Input','Enter an integer','0'));
3. Smallest := 0;
4. For count := 0 to 8 do
Begin
5. Value := strtoint(inputbox('Input','Enter next integer','0'));
6. If value < smallest then smallest := value
End;
7. Lbloutput.caption := inttostr(smallest);
	Scanner sc=new Scanner(System.in);
	1. Int value;
	System.out.println(" Enter an integer: ");
	2. int value = sc.nextInt();
	3. int smallest = 0;
	4. for (int count = 0; count < 8; count++)
	{
	System.out.println(" Enter next integer: ");value = sc.nextInt();
	6. if (value < smallest) smallest = value
	}
	7. System.out.printf("Lowest value is %d ", value);

Answer:

Line	Value	Count	Value<smallest	Smallest

LESSON 7 WORKSHEET
FUNCTIONS AND PROCEDURES I

Lesson Outcomes
- Apply the encapsulation principles in programming
- Show how to pass parameters in methods and use correct return types
- Write methods with encapsulation

Give one word for each of the following:
Information hiding so that other classes have to use accessors and mutators to see or modify information in the class.

_____

A method to make private variables in a class available.

_____

A method used to modify private variables in a class.

_____

Members that can be accessed anywhere where the class can be referenced.

_____

Members that are invisible outside of the program or unit where its class is declared.

_____

Special method to initialise a new instance of a class.

_____

A Value received by a method of a class to change the value of a field of a class.

_____

A data type indicating the data format to be returned by a method.

_____

TASK

1. Study the class diagram below representing a client:

Client	
Fields	**Methods**
- fID: String - fName: String - fDateOfBirth: String - fAmount : double	+ constructor Create/member(ID,name) + getID(): String +getName(): String +getAmount() :double +setAmount(newAmount)

a. Identify a constructor in the class diagram above _____

b. Identify a mutator in the class diagram above _____

c. Identify an accessor in the class diagram above _____

d. Identify a parameter in the example _____

_____

2. Code the accessor method getID:

DELPHI	JAVA

3. Code the mutator setAmount:

DELPHI	JAVA

LESSON 8 WORKSHEET

FUNCTIONS AND PROCEDURES II

Lesson Outcomes

- Understand the meaning and importance of modularity in programming
- Understand the concept of variable scope in programming
- Distinguish between value and reference parameters

Complete the sentences:

_____ is a programming method in which the source code for an object can be written and maintained independently of the source code for other objects.

_____ allows one object to send data to another object or ask the object to invoke a method.

The detail of what goes on inside the module is not important to the system as a whole, as long as the module fulfils its role correctly. This is called _____.

_____ are accessible to all members in the class (can be used in all event handlers).

_____ are declared in methods or code blocks and their scope is limited to that method.

_____ can only be accessed in the method body.

TASK

Consider the following code, keeping the scope of variable in mind. **Predict the output.**

DELPHI

```
Var
 x,z : integer;
procedure TForm1.FormCreate(Sender: TObject);
begin
x:= 4;
end;

function Tform1.calc(z: integer) : integer;
begin
result := z*x ;
end;

procedure TForm1.Button1Click(Sender: TObject);
var x : integer;
begin
x := 5;
z := 2;
RichEdit1.Lines.add('x= '+inttostr(x) +#13
+'z= '+ inttostr(z)+#13);
+'zx= '+inttostr(calc(x))+#13
end;
```

JAVA

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class SwingApplication implements
ActionListener {

   int x = 4;
   int z;
   JLabel outputLabel = new JLabel("Click the button
for output");

   public Component createComponents()
   {
      JButton button1 = new JButton("button1");
      button1.addActionListener(this);
      outputLabel.setLabelFor(button1);
   }
   public void actionPerformed(ActionEvent e)
   {
      x = 5;
      z = 2;
      label.setText("x= " + inttostr(x) +
                  "\tz= " + inttostr(z) +
                  "\tzx= " + inttostr(calc(x)));
   }
   public int calc (int z)
   {
      ans = z*x;
      return ans;
   }
```

LESSON 9 WORKSHEET
STRING MANIPULATION

Name 3 methods to manipulate a string.

_____

_____

_____

Lesson Outcomes

- Recognise the string classes and be able to construct strings
- Write code using string methods for string manipulations and comparison
- Manipulate strings by replacing characters, converting to upper and lower case, performing concatenation and removing spaces

Complete the code to:

	DELPHI	**JAVA**
Convert the string called sentence to uppercase and assign it to variable new		
Convert the string called sentence to lowercase and assign it to variable new		
Copy the first three characters from the string sentence and assign it to variable new		
Assign the length of the string sentence to length		
Assign the 4^{th} character in the string sentence to new		
Assign the position of the first space in sentence to place		

TASK
Create a flowchart for the program to remove the spaces in a sentence.

LESSON 10 WORKSHEET
ARRAYS I

Lesson Outcomes
- Declare an array, set its size and populate the array
- Write code to search through an array
- Create and use a constants array

Write down the declaration for a constants array called arrSuits, to display the four suits in a deck of cards.

_____

What will be assigned to the variable called output in the following cases:

DELPHI	**JAVA**	
Output := arrsuits [0]	Output = arrSuits[0]	
Output := arrSuits[1+2]	Output = arrSuits[1+2]	
Output := arrSuits[1]+arrsuits[2]	Output = arrSuits[1]+arrsuits[2]	

TASK

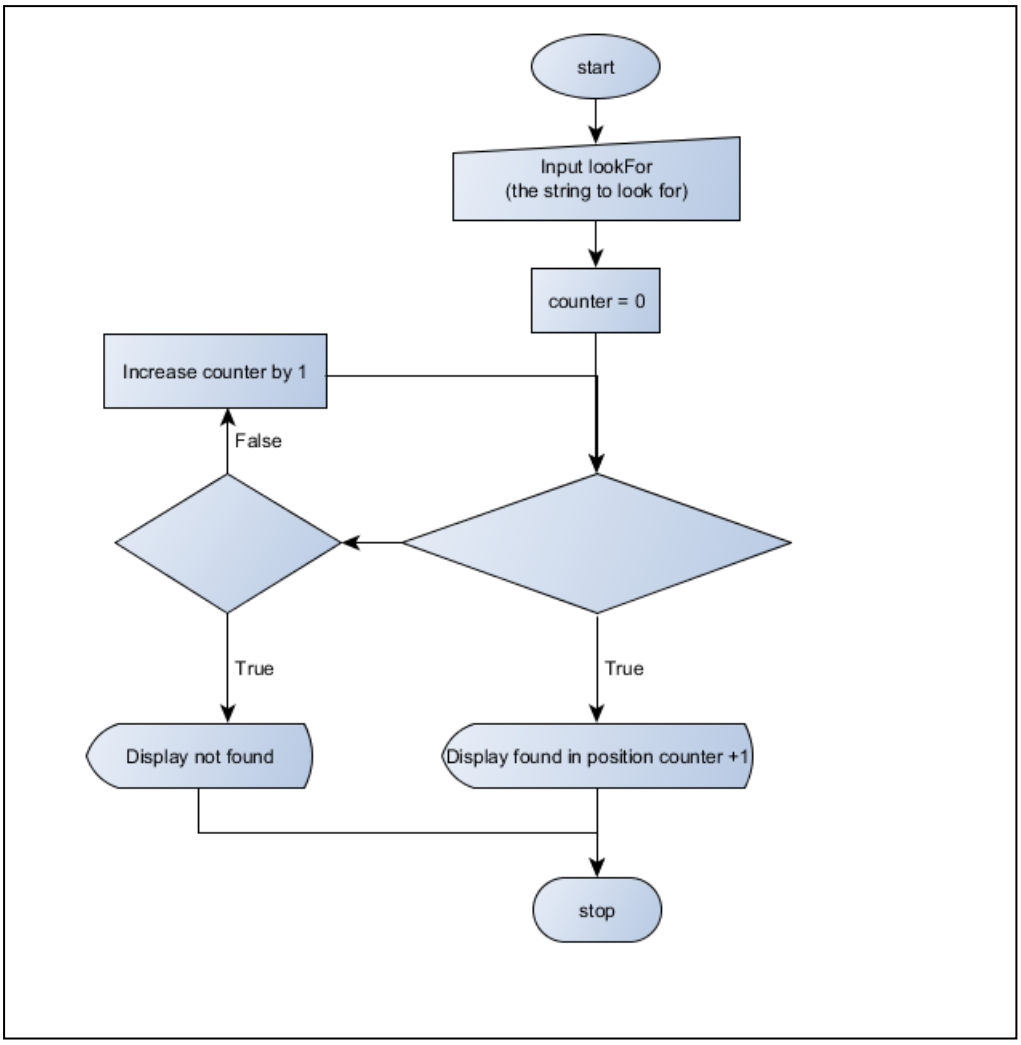
1. Write a loop to populate an array of length 10 with random integers between 20 and 30.

DELPHI	**JAVA**

2. Next, display the integers in an array of length 10 with random integers.

DELPHI	**JAVA**

3. Complete the empty boxes in the flowchart that is supposed to search through an array of 100 names. The search must stop as soon as you find a match or when you reach the end of the array.


start
Input lookFor
(the string to look for)
counter = 0
Increase counter by 1
False
True
True
Display not found
Display found in position counter +1
stop

LESSON 11 WORKSHEET
ARRAYS II

Lesson Outcomes
- Use methods to calculate totals for array items, rows and columns
- Write code to delete the array elements
- Use methods to sort elements of an array

Complete the sentences:
_____ works by swapping two adjacent elements if they are not in order. The process repeats until no more swapping is necessary.
_____ finds elements that are out of order and puts these elements in the right position.
_____ finds the largest/smallest element in the array and puts it in the correct position. It then repeats the process with the remaining part of the array until all elements are in the right order.

Explain why you need a temporary variable when you swap two values in an array.

_____

TASK

1. Declare an array of integer with 4 rows and 5 columns (An array with 4 arrays of 5 integers).

DELPHI	JAVA

2. Write the code to calculate the sum of all the numbers in the 2D array declared in 1.

DELPHI	JAVA

3. Complete the code to delete all items in column 3 of the array (set the values to 0).

DELPHI	JAVA

4. Use a method to sort the elements in the array arr that has n integer elements.

DELPHI	JAVA

LESSON 12 WORKSHEET
ARRAYS III

Lesson Outcomes

- Identify the basic concepts of a two dimensional array
- Show ability to declare and populate a two dimensional array
- Illustrate ability to iterate through a two dimensional array to list array elements

Comment on the types of loops that are used to populate and display 2D arrays.

_____

_____

List 5 examples where 2D arrays could be used in programs.

_____

_____

_____

_____

Explain what is meant by the following declaration:

DELPHI	**JAVA**
Var arr : array[0..3,0..4] of integer;	int[ ][ ] arr = new int[4][5];

Complete the code to populate a 2D array with random integers between 1 and 10.

DELPHI	**JAVA**

TASK

1. Write the code to print a 2D array int[ ][ ] arr = new int[4][5] in rectangular form:

DELPHI	**JAVA**

LESSON 13 WORKSHEET
CLASSES AND OBJECTS I

Lesson Outcomes
- Define and instantiate a class and declare and initialise instance variables
- Explain how to access data members and change data for fields
- Compare class instances and objects

Define a class _____

_____

_____

Define an object _____

_____

_____

True or false? Many objects can be created using the same class. _____

TASK

1. Define a class Tfood. The type of food is of the type string. Add appropriately named private fields of the type integer to hold the following data:
   Foodtype
   Amount of fat
   Amount of Proteins
   Amount of carbohydrates
   Energy
   (The fat, protein and carbohydrates is in grams per 100g and energy is in kJ)

DELPHI	JAVA

2. Write a public constructor method which accepts values for all five private fields through the parameters to initialise the private fields.

DELPHI	JAVA

3. Write an public accessor method getKJ to make the number of kilojoules available in the main.

DELPHI	JAVA

4. Write an public accessor method getType to make the type of food available in the main class.

DELPHI	JAVA

5. Write an public accessor lowFat to return the boolean value true if the is less than 10.

DELPHI	JAVA

LESSON 14 WORKSHEET
CLASSES AND OBJECTS II

Lesson Outcomes
- Define a class to represent an object
- Create and dynamically populate an array of objects
- Write code to search an array of objects

Can you use arrays as data types for the fields that define the attributes? _____

What will happen if you declare the procedures and functions in the private section?

_____

_____

Assume the following class has been created:

DELPHI	**JAVA**
Food = class(Tobject) Private fType : string; fFat,fProtein,fCarbo, fEnergy : integer;	Public class Food { Private string type Private int fat; Private int protein; Private int carbo; Private int energy;

Declare an array named arrFood that can hold 25 objects of the class Food.

DELPHI	**JAVA**

Create and dynamically populate the array of objects with random values between 0 and 100 for all the integer parameters and the names of the food “A”, “B” .. “y”

DELPHI	**JAVA**

TASK

1. Write the code to search through the array of objects to find the food with the name entered by the user.

DELPHI	JAVA

LESSON 15 WORKSHEET
CLASSES AND OBJECTS III

Lesson Outcomes
- Give a detailed explanation of polymorphism and inheritance
- Apply inheritance in classes
- Apply polymorphism in classes

Explain polymorphism with an example.

_____

_____

_____

Name the two types of polymorphism.

_____

Explain inheritance with an example.

_____

_____

TASK

1. Give one example of where polymorphism can be used.

_____

_____

2. Fill in the missing words:

You don't have to write code in _____, you actually just know you will need a method, but the _____ class will determine exactly what the content of the method will be.

Put only the _____ in parent – not specific –, but every child class MUST have a method with exactly the same _____ where the method is specified

.

LESSON 16 WORKSHEET
TEXT FILES

Lesson Outcomes

- Identify basic key words for text files and file data access
- Instantiate a text file and write data to a text w\file
- Read multiple lines of text from a text file

Why is it necessary to close files at the end of a procedure?

_____

_____

Text files are used sequentially. What does this mean?

_____

Is it better to use While loop or a repeat Until (Do while) loop to read through the file? Motivate.

_____

_____

TASK

1. Match the columns with the basic keywords for data access to text files:

Function that links the file variable in RAM to the *.txt file on disk		A.	Append(tf)
Opens an existing file for reading. It puts the file pointer at the beginning of the file.		B.	AssignFile(tf,'abc.txt')
Creates and opens a new file. If existing file with the same name existed, the new one will overwrite the old		C.	Eof (tf)
Opens an existing text file for adding on at the end. It puts the file pointer at the end of the file		D.	Fileexists('abc.txt')
Reads a line from the file into variable line and then move the file pointer to the beginning of next line.		E.	Readln(tf,line)
Writes a line and then writes an end-of-line marker to the text file (moves file pointer to the next line)		F.	Reset (tf)
Function that returns true if the file exists and false if it does not exist		G.	Rewrite(tf)
Function that returns true if you have reached the end of the file and false if there are more lines of text (or end of line markers) to read from the file		H.	Writeln(tf,line)

LESSON 17 WORKSHEET
EVALUATING

Name three types of errors that can occur in programming and give an example of each.

_____

_____

_____

Which type of error can be prevented when you use exception handlers?

_____

_____

_____

Lesson Outcomes

- Explain what an exception is
- Describe exception handling techniques
- Use exceptions and handlers
- Define documentation and use good documentation convention
- Explain different kinds of testing, test cases and test plans
- Explain bugs, debuggers and debugging

Describe what you understand as good documentation in programming.

_____

_____

_____

Describe the types of data to include in your test cases.

_____

_____

_____

TASK

1. How do you use the debugger in your software program to step through your program one line at a time?

   _____

   _____

2. How do you set watches in the software program?

   _____

   _____

3. How do you add breakpoints in the software program?

_____

_____

4. List a couple of bugs that you have encountered in programs while stepping through the code.

_____

_____

_____

LESSON 18 WORKSHEET

GRAPHIC USER INTERFACE (GUI) I

Lesson Outcomes

- Identify the Graphical User Interface (GUI) and design principles
- Identify components in Delphi
- Use Netbeans or Delphi and GUI components to develop a data input form

List 4 design principles when creating a graphical user interface.

_____

_____

_____

_____

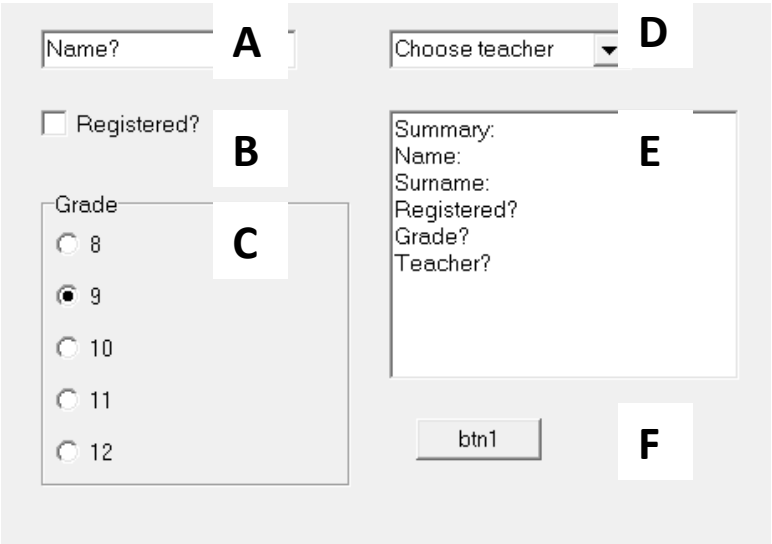
How do you make the interface aesthetically pleasing?

_____

_____

_____

Identify the components on the following GUI:



A: _____

B: _____

C: _____

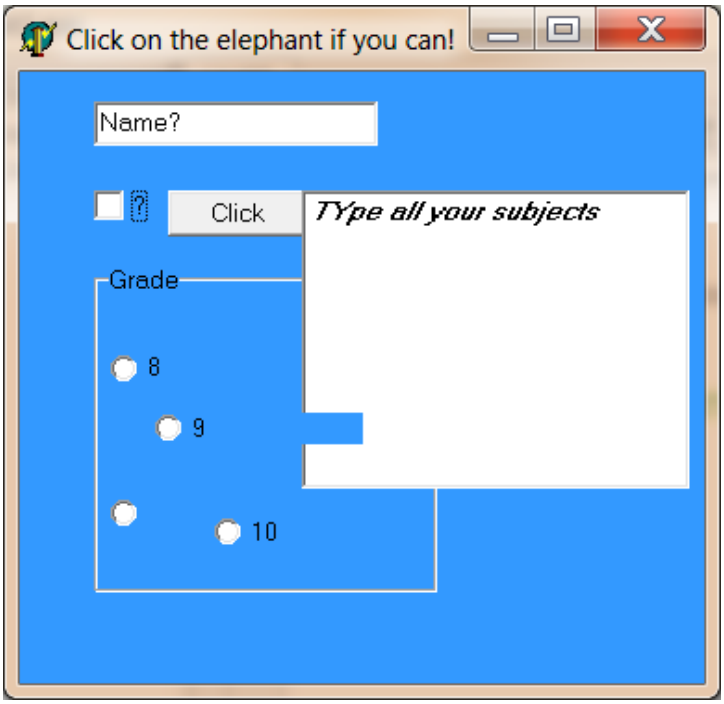
D: _____

E: _____

F: _____

TASK

1. Consider the following user interface and use the design principles to comment on the design:



_____

_____

_____

_____

_____

_____

LESSON 19 WORKSHEET

GRAPHIC USER INTERFACE (GUI) II

Lesson Outcomes

- Show an understanding of the event delegation model in Delphi
- Recognize event classes in Delphi

Name a few events that can be used.

_____

_____

_____

Name 4 types of event listeners.

_____

_____

_____

TASK

1. Choose the appropriate events by matching the columns:

To activate an eventhandler when a button is clicked	FromKeyUp
To change the color of the form when you move the mouse over the panel	FormKeyPress
To capture the letter that you enter on the keyboard to make an object move on the form	OnClick
To be able to use the arrow keys to make an object move on the form	OnMouseMove

2. Complete the code to make an image move on the form when the arrow keys are pressed.

DELPHI	**JAVA**

LESSON 20 WORKSHEET

SQL

Lesson Outcomes

- Explain what SQL is
- Explain how Delphi or JAVA connects to and uses the database
- Write SQL statements in Delphi or JAVA

Write down what the following acronyms stand for:

SQL _____

ODBC _____

JDBC _____

ADO _____

TASK

1. Given the following table in ACCESS, complete SQL statements for the following queries:

Models

ModelID	Model Name	A/C	ABS_brakes	Tank	Seats	Lookup to Groups
1	VW Chico	No	No	42	4	Economy
2	Toyota Yaris T1	Yes	Yes	42	4	Economy
3	Hyundai Atos	Yes	No	45	4	Economy
4	VW Polo Hatch	No	Yes	45	4	Compact
5	Toyota Yaris T3	Yes	Yes	42	4	Compact
6	Toyota Corolla	Yes	Yes	52	4	Intermediate
7	Honda Jazz	Yes	Yes	55	4	Intermediate
8	Audi A3	Yes	Yes	60	5	Intermediate
9	Toyota Corolla	Yes	Yes	52	4	Intermediate
10	New Golf	Yes	Yes	55	4	Intermediate
11	BMW 320	Yes	Yes	60	5	Full Size

i. Display the Model Name and fuel tank size fields from the models table with column headings “Car” and “Fuel tank”.

_____

ii. Display all fields of models where the tank field is blank (empty).

_____

iii. Display all the fields from the models table WHERE the model name starts with A to H .

_____

iv. Display all the fields from the models table WHERE the model description is longer than 10 characters.

_____

v. Display all fuel tank capacities only once.

_____

vi. Display Model names starting with an H.

vii. Write a query where the user can input the number of seats and then all models with that number of seats are displayed.

viii. Display the model names sorted in descending order by seats and then ascending order by fuel tank capacity.

ix. Display the maximum range of all cars in the models table.

x. Display the number of models in the models table.

xi. Change the Air_bags field for the Nissan 4x4 double cab (ModelID = 18) to True.

xii. DELETE records from the rental table WHERE the car registration is VBN307KZN.

xiii. Display the number of cars in each group listed in the groups table, according to the rentals table.

xiv. Insert a new car into the Models table.

ModelID	Model Name	A/C	ABS_brakes	Tank	Seats	GroupID
25	Opel Corsa	Yes	No	45	4	Economy

LESSON 1 WORKSHEET ANSWERS

To write a program to provide information about trees in the botanical garden, you need to declare some variables. Choose the most appropriate data type for the variables. Then choose variable names for these variables, declare the variables and assign the value *Acacia xanthophloea* to the name, 245 to the number of trees, 21.87 to the Average height and NO or false to the Evergreen variable

Description	Data type	Declaration in Delphi	Declaration in JAVA	Assign the value in Delphi	Assign the value in JAVA
Name of the tree	String	Treename: string;	String treeName	Treename:= 'Acacia xanthophloea';	treeName = "Acacia xanthophloea";
Number of this type of tree in the garden	Integer	Number: integer;	Int number	Number := 245;	Number = 245
Average height for this type of tree in meters	Floating point data type/real	AvgHeight: real;	Float avgHeight	AvgHeight:= 21.87;	avgHeight = 21.87;
Evergreen or not	Boolean	Evergreen: boolean;	Boolean evergreen	Evergreen := false;	evergreen = false;

Note: Name is a reserved word in Delphi and should not be used as a variable name
What is the biggest value that can be assigned to a variable of the type **byte**? 255
What happens if you assign a bigger value to the variable than what the data type allows?
You get overflow and might get negative values that are incorrect

TASK ANSWERS

1. Find the errors in the following code:

DELPHI	Corrected:	JAVA	Corrected:

Var
x,y : integer;
z : integer;
begin
x : 5;
y = 6;
z := x/y;
lbloutput.caption := z;
end;
	Var
	x,y : integer;
	z : real;
	begin
	x := 5;
	y := 6;
	z := x/y;
	lbloutput.caption := floattostr(z);
	end;
		Public static void main(string[] args) {
		int x;
		x = 5
		int y;
		y :=6;
		int z = x/y;
		system.out.println(z);
			Public static void main(string[] args) {
			int x;
			x = 5;
			int y;
			y = 6;
			float z;
			z = x/y;
			system.out.println(z);

Note: in Delphi you have to convert the answer z to string to be able to display it. The function floattostr() converts real numbers to string and inttostr() converts integer to string

LESSON 2 WORKSHEET ANSWERS

Fill in the appropriate relational operators for the expressions to be true:

Expression	Answers
8 10	< or <> or !=
'Joe' ….'Jack'	< or <> or !=
'A' 'a'	< or <> or !=

Note: Letters and strings are ordered according to their ASCII values
What is the order of precedence when using boolean and relational operators? NOT AND OR followed by the relational operators

Determine the value of **outcome** in the following cases if C = 10 and D = 20:

DELPHI	**JAVA**	
Outcome := (C<20) and (D<=20)	outcome = (C<20) && (D<=20)	TRUE
Outcome := NOT(C<D)	Outcome = !(C<D_	FALSE
Outcome := (C>=20) or (D<=20)	outcome = (C>=20) OR (D<=20)	TRUE
Outcome := (C>20) OR (D<300) AND NOT(C>D)	outcome = (C>20) OR (D<30)&&!(C<D)	TRUE

TASK ANSWERS

1. Use nested if statements to write a program to display the output "Distinction" if the mark is greater or equal to 80 and less than or equal to 100, "Pass" if the mark is between 40 and 79 (inclusive), "Fail" if the mark is between 0 and 39 (inclusive) and "Invalid" if the mark is > 100 or <0. Use 77 as input for the mark

DELPHI	**JAVA**

Var
Mark : integer;
begin
mark := 77;
if (mark > 100) OR (mark <0) then
lbloutput.caption := "Invalid"
else if mark >= 80 and =<100 then
lbloutput.caption := "Distinction"
else if (mark>= 40) and =<100 then
lbloutput.caption := "Pass"
else lbloutput.caption := "Fail"
	int mark = 77;
	if (mark >= 100) OR(mark <=0){
	system.out.println("Invalid");
	} else if (mark >= 80) {
	system.out.println("Distinction")
	}else if {
	(mark>= 40)
	{system.out.println("Pass")
	}else {
	system.out.println("Fail")
	}

Note: note how you logically exclude values every time you enter the next nested loop

2. Use a case statement or switch to display the season when the number of the month is entered: (Spring: 9,10,11; Summer: 12,1,2;Autumn: 3,4,5; Winter: 6,7,8). Use an integer value month with value 3 in your answer.

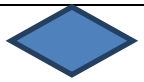
DELPHI	**JAVA**

Var
month : integer;
begin
month := 3;
case month of
9..11: lbloutput.caption := "Spring"
12,1,2: lbloutput.caption := "Summer"
3..5: lbloutput.caption := "Autumn";
6..8:"Winter"
Else lbloutput.caption := "Invalid"
End;
End;
	int month = 3;
	switch (month) {
	case 9: case 10: case 11:
	System.out.println("Spring");
	break;
	case 12: case 1: case 2:
	System.out.println("Summer.");
	break;
	case 3: case 4: case 5:
	System.out.println("Autumn");
	break;
	case 6: case 7: case 8:
	System.out.println("Winter");
	break;
	default:
	System.out.println("Invalid month.");
	break;
	}

Note that you can use sub ranges in DELPHI but you have to list all cases in JAVA

LESSON 3 WORKSHEET ANSWERS

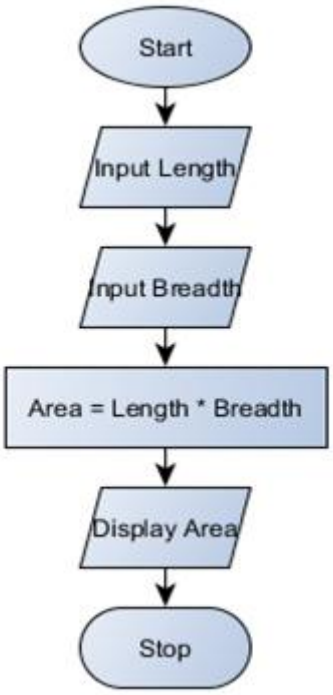
Match the columns: Which symbols are used in flowcharts to indicate the following:

INPUT and OUTPUT	C	A 
DECISIONS	A	B 
PROCESSING	B	C 

What is the purpose of the arrows in flowcharts? To indicate the flow of the program

TASK ANSWERS

1. Use a sequential flowchart to calculate the area of a rectangle.



2. Use pseudocode to plan the solution of finding the largest value of two values that were entered and display “equal” if they are the same.

```
Input x
Input y
If x > y then display x
Else if y > x then display y
Else x = y
```

3. To swap two values using an extra data container

LESSON 4 WORKSHEET ANSWERS

Name 3 types of loops? For, while, repeat(do while)

Name the three parts of a for loop or the three thing the for statement does. Initialise, test (boolean expression) and Change (update) the loop variable

Name the most appropriate loop:

Description	Loop type
To display “Welcome” 10 times	For
To guess a number between 1 and 100 until you get it right	Repeat(Do while)/while
To search though a list until you find a value or reach the end of the list	While
To keep increasing a value by 10% if it is smaller than 50	Repeat(Do while)/while

TASK ANSWERS

Complete the code for the following programs:

1. To display “Welcome” 10 times

DELPHI	JAVA

For I := 1 to 10 do
Redout.lines.add(‘welcome’);
	Int I;
	For (i=1;i<=11;i++){
	system.out.println(“welcome”);
	}

2. To guess a number between 1 and 100 until you get it right

DELPHI	JAVA

Var
Secretnum, guess : integer;
begin
secretnum := random(100)+1;
repeat
guess := strtoint(inputbox(‘Input’;‘Enter your answer’,’50’))
until (guess = secretnum);
	int randomNumber = (int)(Math.random()*100)+1;
	Scanner sc=new Scanner(System.in);
	do {
	System.out.println("Please input an integer");
	Int x;
	x=sc.nextInt();
	while (x!=randomNumber)
	}

3. To keep increasing a value by 10% if it is smaller than 50

DELPHI	JAVA

Var
x : real;
begin
x := 5;
while x <50 do
x := x*1.1;
	float x;
	while (x <50){
	x=x*1.1
	}

LESSON 5 WORKSHEET ANSWERS

INPUT	PROCESSING	OUTPUT	STORAGE
What is the data type of the input? What kind of data does the input represent?	How can the answer be calculated? What basic operators are needed? What is the formula I need? Do I need a loop? Do I need if statements?	What is the data type of the output? How is the answer to be displayed? Do I have to print or display the data or store in permanently? Do I need to format the output?	Where must the data be stored? Do I have to print or display the data or store in permanently?

TASK ANSWERS

1. Complete IPO tables for the following
   a. Read the radius of a circle. Then determine and display the area of the circle

I	P	O
Input radius	Area = pi*radius*radius	Display area

   b. Read the length of three sides of a triangle. Determine if it is a right angled triangle and display a suitable message

I	P	O
Input x,y,z	If x*x=y*y+Z*z or y*y=x*x+z*z or z*z=x*x+ y*y then	Display right angled triangle
	else	Display not right angled triangle

   c. Read a cell phone number. Determine if it starts with a 0 and is 10 characters long to determine if it is a valid cell number and display a suitable message

I	P	O
Input cell	If cell[1] = '0' and length(cell)= 10 then	Display valid format
	else	Display not valid format

2. Implement the IPO tables in the previous question
   a) Radius := strtoint(inputbox('Input','Enter the radius','5'));
   Area := pi*radius*radius
   Lbloutput.caption := floattostr(area)

   b) x := strtoint(inputbox('Input','Enter the length of side 1','5'));
   y := strtoint(inputbox('Input','Enter the length of side 2','3'));
   z := strtoint(inputbox('Input','Enter the length of side 3','4'));
   If x*x=y*y+Z*z
   or y*y=x*x+z*z
   or z*z=x*x+ y*y then
   Lbloutput.caption := 'right angled triangle ' else
   Lbloutput.caption := 'not right angled triangle ' else

   c) cell:= := strtoint(inputbox('Input','Enter cell number-no spaces','0823456789'));
   If cell[1] = '0' and length(cell)= 10 then
   Lbloutput.caption := 'valid format' else
   Lbloutput.caption := 'not valid format'

LESSON 6 WORKSHEET ANSWERS

What is the purpose of the program planned in the flowchart? Loop until the user inputs a value between 8 and 12

What does UML stand for? Unified Modelling Language

TASK ANSWERS

1. Identify possible actors, communication and use cases in the following scenario:

A client brings a pet to the vet. The Vet examines the pet and prescribes medicine or performs a procedure if necessary. The client pays.

ACTORS	USE CASE
client	Makes appointment Buys food, medicine etc Pays account owns pet
Vet	Diagnose Prescribes Performs procedure Invoice client
Pet	Owned by client

2. The following program is supposed to input 10 values in a loop and find the smallest value entered. The output is incorrect. Use a trace table to show how the values of all variables in the program change as each line of code is executed and find the mistake. Explain what to do to debug the program

DELPHI	JAVA

8. Var smallest, value, count : integer;
Begin
9. Value :=
strtoint(inputbox('Input','Enter an integer','0'));
10. Smallest := 0;
11. For count := 0 to 8 do
Begin
12. Value :=
strtoint(inputbox('Input','Enter next integer','0'));
13. If value < smallest then smallest := value
End;
14. Lbloutput.caption :=
inttostr(smallest);
	Scanner sc=new Scanner(System.in);
	1. Int value;
	System.out.println(" Enter an integer: ");
	2. int value = sc.nextInt();
	3. int smallest = 0;
	4. for (int count = 0; count < 8; count++)
	{
	5. System.out.println(" Enter next integer: ");
	value = sc.nextInt();
	6. if (value < smallest) smallest = value
	}
	7. System.out.printf("Lowest value is %d ", value);

Line	Value	Count	Value < smallest	Smallest
2	7	?	F	?
3	7	?	F	0
4	7	0	F	0
5	9		F	0
			F	
			F	

Set smallest to the first value entered instead of 0 otherwise it will only change when you enter negative values

LESSON 7 WORKSHEET ANSWERS

Give one word for the following:

Information hiding so that other classes have to use accessors and mutators to see or modify information in the class	Encapsulation
A method to make private variables in a class available	Accessor
A method used to modify private variables in a class	Mutator
Members that can be accessed anywhere where the class can be referenced	Public member
Members that are invisible outside of the program or unit where its class is declared	Private member
Special method to initialise a new instance of a class	Constructor
A Value received by a method of a class to change the vlaue of a field of a class	Parameter
A data type indicating the data format to be returned by a method	Return type

TASK ANSWERS

1. Study the class diagram below representing a client:

Client	
Fields	**Methods**
- fID: String - fName: String - fDateOfBirth: String - fAmount : double	+ constructor Create/member(ID,name) + getID(): String +getName(): String +getAmount() :double +setAmount(newAmount)

a. Identify a constructor in the class diagram above + constructor Create/member(ID,name)

b. Identify a mutator in the class diagram above +setAmount(newAmount)

c. Identify an accessor in the class diagram above + getID(): String
+getName(): String
+getAmount() :double

d. Identify a parameter in the example new amount

2. Code the accessor method getID

DELPHI	**JAVA**

Function getID : string;
Begin
Return: fID;
End;
	Public String getID(){
	Return fID
	}

3. Code the mutator setAmount

DELPHI	**JAVA**

Procedure setAmount(newAmount:double);
Begin
fAmount := newAmount
End;
	Public void setAmount(double newAmount){
	fAmount = newAmount
	}

LESSON 8 WORKSHEET ANSWERS

Complete the sentences:

Modularity is a programming method in which the source code for an object can be written and maintained independently of the source code for other objects.
Modularity allows one object to send data to another object or ask the object to invoke a method
The detail of what goes on inside the module is not important to the system as a whole, as long as the module fulfils its role correctly. This is called information hiding
Global variables are accessible to all members in the class (can be used in all event handlers)
Local variables are declared in methods or code blocks and their scope is limited to that method
Method parameters can only be accessed in the method body

TASK ANSWERS

1. Consider the code (provided), keeping the scope of variable in mind. Predict the output:

Output:

X = 5 the local value of x
Z = 2 the global value of z
Zx = 20 local x = 5 passed to parameter z and in function the global value of x (4) is used to multiply

LESSON 9 WORKSHEET ANSWERS

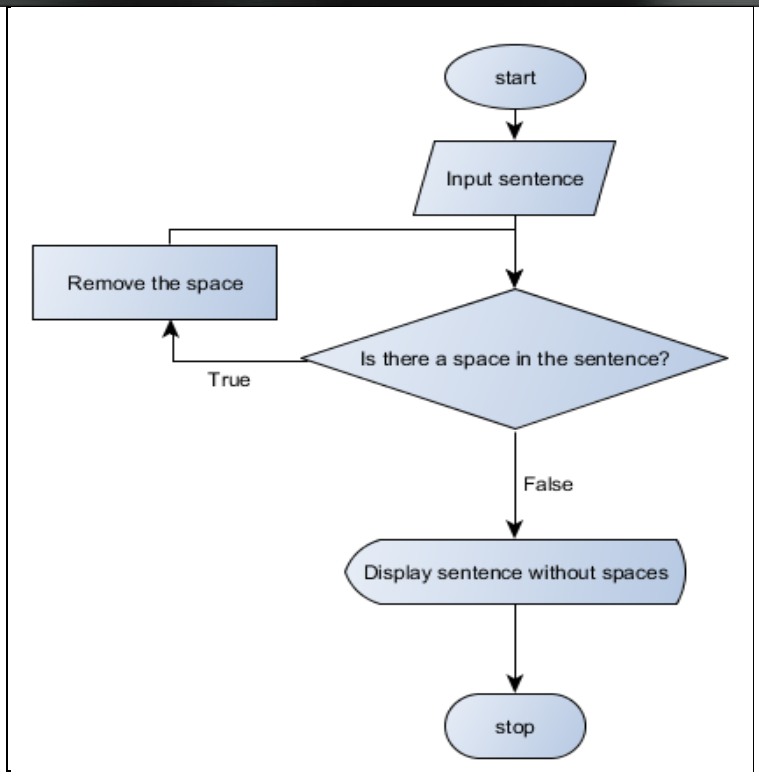
Name 3 methods to manipulate a string: A method on the string type is called to manipulate the string. It can be used to find the length of a string. It can be used to count the number of times a given word or character appears in a string. It can be used to convert lowercase into uppercase. It can be used to reverse the order of characters, e.g. abcd to dcba.

Complete the code to:

	DELPHI	JAVA
Convert the string called sentence to uppercase and assign it to variable new	New := uppercase(sentence)	String new = sentence.toUpperCase();
Convert the string called sentence to lowercase and assign it to variable new	New := lowercase(sentence)	String new = sentence.toLowerCase();
Copy the first threee characters fromm the string sentence and assign it to variable new	New := copy(sentence,1,3)	String New = sentence.substring(0,3)
Assign the length of the string sentence to len	Len := length(sentence);	Int len = sentence.length();
Assign the 4th character in the string sentence to new	New := sentence[4]	Char new := sentence.charAt(3)
Assign the position of the first space in sentence to place	Place := pos(' ',sentence)	Int place = sentence.indexOf(' ')

TASK ANSWERS

1. Create a flowchart for the program to remove the spaces in a sentence.



LESSON 10 WORKSHEET ANSWERS

Write down the declaration for a constants array called arrSuits, to display the four suits in a deck of cards

JAVA String[ ] arrSuits = {'Hearts','Spades','Clubs','Diamonds'};

DELPHI arrSuits = array[0..3] of string = ('Hearts','Spades','Clubs','Diamonds');

What will be assigned to the variable called output in the following cases:

DELPHI	JAVA	
Output := arrsuits [0]	Output = arrSuits[0]	Hearts
Output := arrSuits[1+2]	Output = arrSuits[1+2]	Diamonds
Output := arrSuits[1]+arrsuits[2]	Output = arrSuits[1]+arrsuits[2]	SpadesClubs

TASK ANSWERS

1. Write a loop to populate an array of length 10 with random integers between 20 and 30.

DELPHI	JAVA

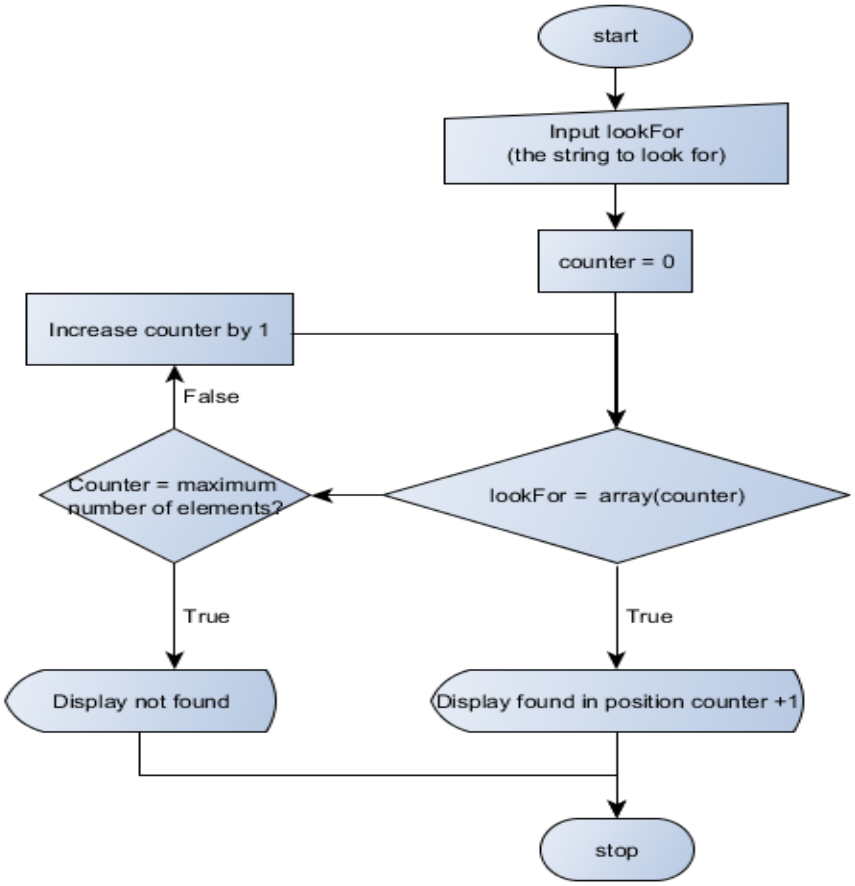
Var arr : array[0..9] of integer;
k : integer;
Begin
Randomize;
For k := 0 to 9 do
Arr[k] := random(11)+20
	int arr[]=new int[10];
	
	for(int i=0;i<10;i++){
	arr[i]= Math.round(Math.random() * (10)) + 20;
	}

2. Display the integers

DELPHI	JAVA

Var arr : array[0..9] of integer;
k : integer;
Begin
For k := 0 to 9 do
Redoutput,lines.add(inttostr(Arr[k]))
	System.out.println("Random numbers are: ");
	for(int i=0;i<random.length;i++){
	System.out.println(random[i]);
	}
	}
	}

3. Complete the empty boxes in the flowchart that is supposed to search through an array of 100 names. The search must stop as soon as you find a match or when you reach the end of the array



LESSON 11 WORKSHEET ANSWERS

Complete the sentences:

a. Bubble sort works by swapping two adjacent elements if they are not in order. The process repeats until no more swapping is necessary
b. Insertion sort finds elements that are out of order and puts these elements in the right position
c. Selection sort finds the largest/smallest element in the array and puts it in the correct position. It then repeats the process with the remaining part of te array until all elements are in the right order

Explain why you need a temporary variable when you swap two values in an array.

Otherwise you will overwrite one of the values and in effect duplicate one value

TASK ANSWERS

1. Declare an array of integer with 4 rows and 5 columns (An array with 4 arrays of 5 integer)

DELPHI	JAVA
Var arr : array[0..3,0..4] of integer;	int[ ][ ] arr = new int[4][5];

2. Write the code to calculate the sum of all the numbers in the 2D array declared in 1

DELPHI	JAVA

Sum := 0;
For col := 0 to 4 do
For row := 0 to 3 do
Sum := sum + Arr[row,col] ;
Showmessage('Sum'+ inttostr(sum))
	// Sum all elements
	Int Sum = 0;
	for (int col=0; col<arr.length; col++) {
	for (int row=0; row<arr[col].length; row++)
	{
	sum = sum + arr[col][row];
	System.out.print("SUM " +sum);

3. Complete the code to delete all items in column 3 of the array (set the values to 0)

DELPHI	JAVA

For row := 0 to 3 do
Arr[row,3] := 0;
	for (int row=0; row<arr[col].length; row++) {
	arr[3][row] := 0;
	}

4. Use a method to sort the elements in the array arr that has n integer elements

DELPHI	JAVA

Procedure bubble_sort(var arr: Tarr;n: integer);	Public static void bubble_sort(int arr[], int n){
	Int I,j,keep = 0;
	
	for (i=0; I < n; i++) {
	for (j=0; j< n-1; j++) {
	if (arr[i] < arr[j] {
	keep = arr[i];
	arr[i] = arr[j];
	arr[j] = keep
	}
	}
	}

LESSON 12 WORKSHEET ANSWERS

Comment on the types of loops that are used to populate and display 2D arrays. Use two For Loops, nested one inside the other when creating a 2D array. The outer For Loop – rows; inner For Loop – columns.

List 5 examples where 2D arrays could be used in programs.

An array variable can be used to store many data items of the same kind under the same name. Each item is distinguished from the other by means of an index. Examples:

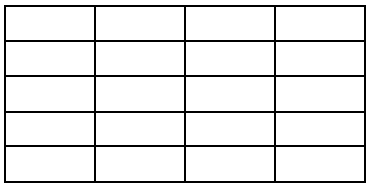
1. A program to book seats for a movie
2. A program that can do the times table and display it in a string grid
3. Useful to store objects for programming sketches that involve some sort of grid or matrix
4. A program that lets a user enter some values at the beginning of each day, such as the number of appointments

5. A program that contains to check if specific items or seats or similar are available, for example, to allow people to book the seat they want on a bus or at a music festival.

Explain what is meant by the following declaration:

DELPHI	JAVA
Var arr : array[0..3,0..4] of integer;	int[ ][ ] arr = new int[4][5];

DELPHI:
It declares a 2D array of integers with 4 columns and 5 rows
JAVA:
It declares a variable, arr of type int[ ][ ], and it initializes that variable to refer to a newly created object. That object is an array of arrays of ints. Here, the notation int[4][5] indicates that there are 4 arrays of ints in the array A, and that there are 5 ints in each of those arrays.



Complete the code to populate a 2D array with random integers between 1 and 10.

DELPHI	JAVA

For col := 0 to 3 do
For row := 0 to 4 do
Arr[col,row] := random(10) +1;
	for(int col=0;col<4;col++){
	for int row= 0;row<3;row++{
	arr[col][row]=
	Math.round(Math.random()*(10))+1;
	}
	}

TASK ANSWERS

1. Write the code to print a 2D array int[ ][ ] arr = new int[4][5] in rectangular form:

DELPHI	JAVA

Sum := 0;
For row := 0 to 4 do
Begin
For col := 0 to 3 do
Line := line + inttostr(Arr[row,col]);
Redout.lines.add(line) ;
End;
	for (int row=0; row<arr.length; row++) {
	for (int col=0; rcol<arr[row].length; row++) {
	System.out.print(" " + arr[col][row]);
	}
	System.out.println("");

LESSON 13 WORKSHEET ANSWERS

Define a class. A class is a blueprint for an object The class groups together values that logically belong together in a structure like a new data type. A class is made up of fields (private variables that contain data values) and methods (procedures and functions). The class is not a variable – you need to declare an object to use

Define an object. An object created from a class is called an instance of the class. One object is a bundle of variables and related functions and procedures

True or false: Many objects can be created using the same class. True. Many objects can be created from the same class, just like many variables can be declared of the same data type.

TASK ANSWERS

1. Define a class Tfood. The type of food is of the type string. (other information also provided)

DELPHI	JAVA
Food = class(Tobject) Private fType : string; fFat,fProtein,fCarbo, fEnergy : integer;	Public class Food { Private string type Private int fat; Private int protein; Private int carbo; Private int energy;

2. Write a public constructor method which accepts values for all five private fields through the parameters to initialise the private fields

DELPHI	JAVA

constructor create(t,f,p,c,e: integer);
begin
ftype := t;
fFat := f;
fProtein := p;
fCarbo := c;
fEnergy := e;
end;
	Public Food String t, int f,int p,int c, int e
	{
	Type = t;
	Fat = f;
	Protein = p;
	carbo = c;
	energy = e;
	}

3. Write an public accessor method getKJ to make the number of kilojoules available in the main class

DELPHI	JAVA

function getKJ : integer;
begin
result := fenergy
end;
	Public int getKJ()
	{
	Return energy
	}

4. Write an public accessor method getType to make the type of food available in the main class

DELPHI	JAVA

function getKJ : string;
begin
result := ftype
end;
	Public String getKJ()
	{
	Return type
	}

5. Write an public accessor lowFat to return the boolean value true if the is less than 10

DELPHI	JAVA

function lowfat : boolean
begin
result := fFat < 10
end;
	Public boolean lowFat
	{
	Return = fat < 10
	}

LESSON 14 WORKSHEET ANSWERS

Can you use arrays as data types for the fields that define the attributes? Yes

What will happen if you declare the procedures and functions in the private section? Won't be able to use them in any other class (application)

Assume the following class has been created (see information provided in worksheet). Declare an array named arrFood that can hold 25 objects of the class Food

DELPHI	JAVA
Var arrFood: array[1..25] of Food	Food[ ] = arrFood= new Food[25];

Create and dynamically populate the array of objects with random values between 0 and 100 for all the integer parameters and the names of the food "A", "B" .. "y"

DELPHI	JAVA

Var c : char;
Counter : integer;
Begin
Counter := 0;
For c := 'A' to 'Y' do
Begin
arrFood[counter] :=
Food.create(c,random(101),random(101),random(101) , random(101))
	char letter;
	int counter = 0;
	for(letter = 'A'; letter <= 'Y'; letter++){
	int ff = (int)(Math.random()*100)+1;
	int pp = (int)(Math.random()*100)+1;
	int cc = (int)(Math.random()*100)+1;
	int ee = (int)(Math.random()*100)+1;
	counter ++;
	
	arrFood[counter] =
	new Food(letter,ff,pp,cc,ee);
	}

TASK ANSWERS

1. Write the code to search through the array of objects to find the food with the name entered by the user.

DELPHI	**JAVA**

Var i : integer;
Found : boolean;
Searchfood : char;
Begin
I:= 0;
Found:= false;
Searchfood := inputbox('Input','Enter the letter for the foodtype','C')[1];
While (found = false) and (I < 25) do
Begin
Inc(i);
If arrFood[i].gettype= searchfood then
Found := true;
End;
If found then showmessage(arrFood[i].getkj)
else
Showmessgae('Not found');
End;
	int i = 0;
	boolean found = false;
	System.out.println();
	System.out.print("Enter a food : ");
	char searchFood =
	inKb.readLine().toUpperCase();
	while (found = false)&&(i<25){
	i++;
	if(arrFood[i]
	.getType().toUpperCase().equals(searchFood)
	found = true
	}
	If found
	System.out.println(arrFood[i].getKJ);
	Else
	System.out.println("not found");

LESSON 15 WORKSHEET ANSWERS

Explain polymorphism with an example. Polymorphism means the ability to take more than one form. This is something similar to a word having several different meanings depending on the context. It is the important concept of OOP, from the Greek concept 'poly' (means many) and 'morphism' (means form). For example, the same function or function name is used but the behaviour will be different.

Name the two types of polymorphism. Method overloading(static polymorphism...early binding) and method overriding(dynamic polymorphism...late binding).

Explain inheritance with an example. Object-oriented programming allows classes to *inherit* commonly used state and behavior from other classes. For example, Food, in our previous example can now become the superclass of meat and fruit.

TASK ANSWERS

1. Give one example of where polymorphism can be used.

Possible answer: Superclass animal with a move() method and subclasses bird and dog where move() will mean completely different things. Superclass shape where a area method is declared and subclasses Circle, Rectangle where the meaning of area is completedly different.

2. Fill in the missing words:

You don't have to write code in parent - you actually just know you will need a method , but the child class will determine exactly what the content of the method will be.
Put only the signature in parent – not specific –, but every child class MUST have a method with exactly the same signature, where the method is specified.

LESSON 16 WORKSHEET ANSWERS

Why is it necessary to close files at the end of a procedure? CloseFile frees the memory space that was previously reserved for the file. It also sends any data in the file buffer to the file on disk and updates the file's directory details on disk - and, it usually counts 1 mark!!!

Text files are used sequentially. What does this mean? Have to read through all the lines to get to a specific line (like video tape)

Is it better to use While loop or a repeat Until (Do while) loop to read through the file? Motivate
WHILE - take into account the (unlikely) possibility that the file exists but does not contain any data.

TASK ANSWERS

3. M6atch the columns with the basic keywords for data access to text files:

Function that links the file variable in RAM to the *.txt file on disk	B	AssignFile(tf,'abc.txt')
Opens an existing file for reading. It puts the file pointer at the beginning of the file.	F	Reset (tf)
Creates and opens a new file. If existing file with the same name existed, the new one will overwrite the old	G	Rewrite(tf)
Opens an existing text file for adding on at the end. It puts the file pointer at the end of the file	D	Append(tf)

Reads a line from the file into variable line and then move the file pointer to the beginning of next line.	E	Readln(tf,line)
Writes a line and then writes an end-of-line marker to the text file (moves file pointer to the next line)	H	Writeln(tf,line)
Function that returns true if the file exists and false if it does not exist	D	Fileexists('abc.txt')
Function that returns true if you have reached the end of the file and false if there are more lines of text (or end of line markers) to read from the file	C	Eof (tf)

LESSON 17 WORKSHEET ANSWERS

Name three types of errors that can occur in programming and give an example of each.
Compiler errors including syntax errors: e.g. when you do not declare a variable, spell incorrectly, do not convert to the correct data type.
Runtime errors: try to open a file that does not exist, input letters when integer input is expected.
Logical errors: your program runs, but do not deliver the expected output.

Which type of error can be prevented when you use exception handlers? Runtime errors

Describe what you understand as good documentation in programming. Documentation is the written text that accompanies computer software. It either explains how it operates or how to use it. You should comment in your programming code to explain the purpose of each method and provide a user manual for the end users of your program. Technical details like object design and database design should be available for other programmes who want to improve or build on your code.

Describe the types of data to include in your test cases. Test that your program will work correctly with:

1. No data -Generate and display error messages.
2. Valid data –Program should function as per requirements
3. Invalid data set: - check application behaviour for negative values, alphanumeric string inputs and generate and display error messages.
4. Illegal data format: - System should not accept data in invalid or illegal format-generate error messages
5. Boundary /extreme data(very large or very small)
6. Large amounts of data

TASK ANSWERS

1. How do you use the debugger in your software program to step through your program one line at a time. Press F7 repeatedly or use the icon.
2. How do you set watches in the software program? Ctrl-F5 or use the menu
3. How do you add breakpoints in the software program? Use the menu Run, Add breakpoint or click in the left hand side margin.
4. List a couple of bugs that you have encountered in programs while stepping through the code. Loops or selection code does not have brackets or begin and end statements in the correct places. Incorrect conditions in loops. Variables redeclared or out of scope.

LESSON 18 WORKSHEET ANSWERS

List 4 design principles when creating a graphical user interface.

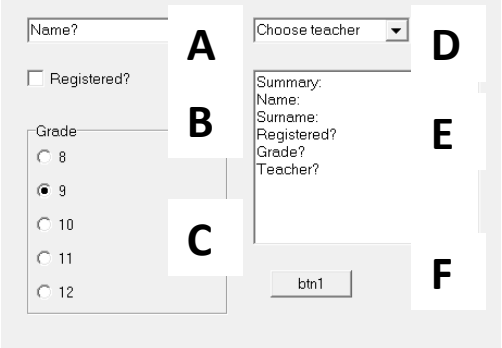
1. Provide visual appeal

2. Clarity - The interface and language used should be easy to understand
3. Consistency - Similar components should have a similar look, position and use
4. Efficiency - Minimize eye and hand movements
(or other meaningful answers)

How do you make the interface aesthetically pleasing?

- Provide meaningful contrast between screen elements.
- Create groupings.
- Align screen elements and groups.
- Provide three dimensional representation
- Use colours and graphics effectively and simply.

Identify the components on the following GI



A: Edit

B: Check box

C: Radio group

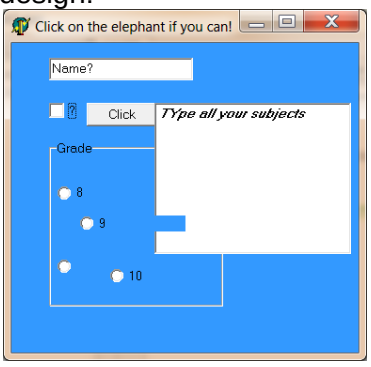
D: Combobox

E: Richedit

F: Button

TASK ANSWERS

Consider the following user interface and use the design principles to comment on the design:



1. Layout not visually appealing – no alignment, overlap, too small and too many components
2. Lack of Clarity - Some components not labelled
3. Inconsistency - Click button should be bottom right if it is the last thing to do
4. In-efficiency - Have to type all subjects – could have selected with check boxes
5. (or other meaningful answers)

LESSON 19 WORKSHEET ANSWERS

Name a few events that can be used. On Click for buttons, Double cllick, Menu item is selected, Mouse events onMousemove, OnCreate, onCLose, OnActivate for windows, OnKeypressed, OnKeyUp, OnkeyDown.

Name 4 types of event listeners. Action listener, MouseListener, MouseMotionListener, WindowListener.

TASK ANSWERS

1. Match the columns: Choose the appropriate events.

To activate an eventhandler when a button is clicked - = OnClick
To change the color of the form when you move the mouse over the panel = OnMouseMove
To capture the letter that you enter on the keyboard to make an object move on the form = FormKeyPress
To be able to use the arrow keys to make an object move on the form = FormKeyUp

2. Complete the code to make an image move on the form when the arrow keys are pressed

DELPHI

```
procedure TForm1.FormKeyUp(Sender:
TObject; var Key: Word;
  Shift: TShiftState);
begin
if key = vk_left then img1.Left := img1.Left -
10;
if key = vk_right then img1.Left := img1.Left +
10;
if key = vk_up then img1.top := img1.top - 10;

if key = vk_down then img1.top := img1.top+
10;
end;
```

JAVA

```
{
        public void keyPressed(KeyEvent ke)
        {
                switch (ke.getKeyCode())
                {
                        case
KeyEvent.VK_RIGHT:
                                x = x + 10;
                        case
KeyEvent.VK_LEFT:
                                x = x - 10;
                        case
KeyEvent.VK_DOWN:
                                y = y - 10;
                        case
KeyEvent.VK_UP:
                                y = y + 10;
                }
                this.repaint();
        }
}
```

LESSON 20 WORKSHEET ANSWERS

Write down what the following acronyms stand for:

SQL Structured Query Language
ODBC Open Database connectivity
JDBC Java Database connectivity
ADO ActiveX data components

TASK ANSWERS

Given the table in ACCESS (see worksheet): Complete SQL statements for the following queries:

i. Display the Model Name and fuel tank size fields from the models table with column headings “Car” and “Fuel tank”
SELECT [Model Name] as Car , tank as [Fuel tank] from models

ii. Display all fields of models where the tank field is blank (empty)
SELECT *FROM models where tank is null

iii. Display all the fields from the models table WHERE the modelname starts with A to H
SELECT * FROM models where [model name] between “A” and “I”

iv. Display all the fields from the models table WHERE the model description is longer than 10 characters
SELECT * FROM models where len([model name]) > 10

v. Display all fuel tank capacities only once
SELECT Distinct tank FROM models

vi. Display Model names starting with an H
SELECT [model name] FROM models where [model name] like “H%”

vii. Write a query where the user can input the number of seats and then all models with that number of seats are displayed
SELECT [model name] from models where seats = '+seats; (where seats is a variable and the user entered the value

viii. Display the model names sorted in descending order by seats and then ascending order by fuel tank capacity
SELECT [model name] FROM models ORDER BY seats desc, tank ';

ix. Display the maximum range of all cars in the models table;
SELECT max(range) as Maximum from models

x. Display the number of models in the models table
SELECT couontrange) as [Number of cars] from models

xi. Change the Air_bags field for the Nissan 4x4 double cab (ModelID = 18) to True.
UPDATE models SET air_bags = true WHERE modelID = 18

xii. DELETE records from the rental table WHERE the car registration is VBN307KZN
DELETE * from rentals WHERE carregistration = "VBN307KZN"

xiii. Display the number of cars in each group listed in the groups table, according to the rentals table
SELECT [model name], sum(date_until - date_from) as [Total days] from models,rentals where models.modelID= rentals.modelID group by [model name]

xiv. Insert a new car into the Models table.

ModelID	Model Name	A/C	ABS_brakes	Tank	Seats	GroupID
25	Opel Corsa	Yes	No	45	4	Economy

INSERT into models (modelID,[model name],[a/c],abs_brakes,tank,seats,groupID) values (25,"Opel Corsa" ,yes, no, 45,4,4,"Economy")'

The Practical Assessment Task

Series overview

There are twelve video lessons in this series which covers phases 1 and 2 of the Practical Assessment Task, and four interactive, PC lessons for phase 3. All practical skills that were studied in Grades 10, 11 and 12 are now put together in a real life situation. In phase 1, the learners learn how to investigate a problem. They evaluate current software programs by questioning the client as well as end-users. Through this process the learner then learns to determine the requirements for their task. During the phase 2 lessons, the learner gets into the depths of designing the software. Learners will learn to understand all the different aspects that must be taken into consideration during the design of a program. Different aspects like input, output, processing and storage must be planned in detail with the use of algorithms, IPO tables, data structures, the database design as well as the user Interface elements. Phase 3 puts all aspects together, implementing analysis and design

Curriculum links

The lessons are designed to cover the following topics as stated in the CAPS document:

- Basic concepts and developing of an algorithm
- Simple flow charts and IPO diagrams to represent an algorithm
- Problem solving techniques
- Planning of a solution
- Principles of Object-Orientated Programming
- Using programming language constructs in the execution of various database transactions
- Design of a program
- Planning techniques and tools.
- Database design
- Defensive programming techniques
- Document a solution design and development
- Design and develop a user defined class to meet the program specifications as part of the solution

Educational approach

The PAT assesses the learner's ability to develop a solution for a specific task. From Grade 10 the concepts of software development were broken down and taught as loose standing skills. In this series we do revision on some of the important concepts of software development while the learner has to put all the separate components together as a programmer should.

A certain scenario is sketched where the learner will have to identify the problem. They must do an investigation to this problem by talking to possible end-users as well as searching for possible current solutions. The users will help the learner to identify the exact requirements of the program regarding the input, processing and output. The result of phase 1 will thus be a proposed solution to the problem. During phase 2 of the PAT, the learner must design the program on paper. With algorithms and IPO tables the learner will design the program. The learner will use a combination of the software development tools studies from Grades 10 – 12 for the solution. The learner should apply appropriate software engineering principles in developing the solution in phase 3. The PAT is basically a revision of all the software developing skills studied.

The Practical Assessment Task

Series at a glance

Lesson title	Lesson Outcomes By the end of this lesson the learner should be able to:
VIDEO LESSONS	
1. Phase 1 - Investigation and Analysis of a problem	• Recognise the need for custom software • Understand the steps involved in custom development • Outline the key aspects • Explore and evaluate educational sub-genres • Identify end-user expectations for this educational software genre
2. Phase 1 - Identifying the requirements	• Identify requirements for: ○ User input from mouse or keyboard ○ Processing like calculations, comparisons, searching and sorting etc. ○ Storage in files on hard drive ○ Output by the program to the monitor • Describe a possible solution
3. Phase 2 - Program Specifications	• Explain the difference between program requirements and program specifications • List expected events for a given program scenario • Classify events according to the sequence they occur in • Break down general statements of events into component parts
4. Phase 2 - Algorithms I	Break an event down into its component steps, clearly showing: ○ sequence ○ selection ○ iteration
5. Phase 2 - Algorithms II	• Explain modularity, identify re-usable modules of code and write algorithms for the modules • Draw a flowchart to illustrate a flow of events
6. Phase 2 - Drawing up IPO Tables	Draw up an IPO table for a given event, to include: ○ Input ○ Processing ○ Output
7. Phase 2 - Storing Data in IPO Tables	• Identify primitive, structured and abstract data types to use for temporary storage • Identify file types and structures to use for permanent storage • Indicate where data validation, error checking and exception handling are required
8. Phase 2 - Classes, Objects and Data Flow I	• Understand classes and objects used for data storage in object orientated programming • Explain the advantages of data encapsulation • Differentiate between private and public members of a class

9. Phase 2 - Classes, Objects and Data Flow II	• Create class diagrams • Create simple data flow diagrams
10. Phase 2 - Database Design I	• Identify data that must be stored permanently • Determine the data that must be stored in text files • Indicate how data structure will be shown in each text file
11. Phase 2 - Database Design II	• Sort data items according to the entity of each item • Determine natural fields and add suitable primary keys • Add foreign fields where relationships between tables must be created • Determine appropriate data types for fields • Prepare a list of queries, reports and table manipulation
12. Phase 2 - User Interface Design	• Illustrate and explain the navigation between screens • Indicate the accessibility of different screens to different categories of user • Illustrate and explain the intended screens fully
INTERACTIVE, COMPUTER LESSONS	
1. Phase 3 - Database Tables	• Analyse data to be stored in order to construct multiple tables • Set appropriate data types for each field in each table and set field properties • Construct lists of lookup values to input data
2. Phase 3 - Database Relationships	• Identify areas where one table in a database needs to reference the primary key field of another table in order to lookup values in records in that table • Identify data that must be stored outside of the database (in text files) • Determine the nature of all relationships • Use database tools to create foreign fields • Establish links between tables in with one-to-many relationships tables • Use database tools to import data • Establish links between tables in with one-to-one relationships • Set parameters in the relationships to maintain referential and data integrity.
3. Phase 3 - GUI Components	• Select appropriate main container components (main screens) to hold groups of other components • Create the necessary links to navigate between the container components • Select appropriate components to store data temporarily during program execution, to provide links to permanently stored data, and to output data on the GUI • Change the default settings of these components to optimise their functionality in a given context
4. Phase 3 - The GUI and the User	• Select appropriate secondary container components (areas on a screen) to group related components. • Select appropriate components to input data from both keyboard and mouse.

	• Change the default settings of these components to optimise their functionality in a given context. • Validate input data where necessary, in order to prevent errors in data manipulation. • Select appropriate components to give information to the user, presented either as static (always visible) or dynamic (context-sensitive) information, in order to guide users to manipulate the data as intended by the program.

Teaching Guidelines – video lessons

Lesson 1: Phase 1- Investigating the Problem

In this lesson we investigate the problem by doing a research and analysis. The learners have to understand that the given scenario is NOT the problem statement. They need to have an interview with the client to find the needs or requirements. Research methods like interviews, surveys, questionnaires can be used to find the needs of end-users. All notes must be kept during this investigation as well as the name and maybe a telephone number of the person. This is useful if you need more information. Most of this investigation is about their current system – its nature, limitations and challenges. Only then the learner can make a decision of the type of program that will be the best, e.g. drill and practice, a game, teaching, etc.

Task

This task is a situation in the IT class where the teacher wants a program to help IT learners with terminology. The teacher wants a program that can help the learners to find all the terms in one place and then to learn them easily.

Notes and Teaching suggestions

The investigation can be time consuming. This task can be done as a group project. Let the learners prepare and ask the IT teacher some questions. Afterwards you can point out the irrelevant questions as well as important questions never asked. The class can also be divided into a few groups. One group set questions for a questionnaire which another group will then fill in as the end-users. Another group can do some research on the internet about available systems. Each group then gives feedback to the class about their section. The model answer can be shared with the learners after their own feedback.

Lesson 2: Phase 1 - Identifying the Requirements

The learners now identify the input, output, processing and storage requirements of the software to help them propose a solution to the problem. By referring to the information they gathered in the investigation they determine the user input which will be either from the keyboard or the mouse. Remember that selecting an option is also input. For each input the learners must decide how it will be processed, where it will be stored and whether there will be any output like a message displayed. To help them to see the program in context it may be better to divide the program in stages (or screens). List the requirements for each stage. This will also give the client a better idea of what the program will be able to do. The possible solution should also include the scope of the program (what it CAN do) as well as the limitations (what it won't be able to do).

Task

A different scenario is used for the task. Learners must identify the requirements for a netball program that must record games, check if all the rules have been followed and calculate the results.

Notes and Teaching suggestions

The investigation should be done like in lesson 1 before the learners can do the program requirements. Either the teacher or a few netball players can find out from the netball organiser what exactly he/she wants to see in such a program. With this information, the class can as a group decide on the different stages that the program may have. Next the class can be divided into groups where each group can take a stage to identify all the requirements for that specific stage. All groups can then get together and with the help of

the teacher compile the final document. As the teacher goes through the work of each group, encourage the learners to discuss if the best options were used for each stage. *Note that this document must be in a format that the client will understand. The groups can make screenshots of the GUI of possible input or output screens to help the client understand the proposal better. Remember to add the scope and limitations in the final product.*

Lesson 3: Phase 2 - Program Specifications

This is the first part that the learners learn how to design the solution for their program. In this lesson they use the program requirements to figure out what the program must do. Learners must make a list of the program events without saying how the event will be done or which components will be used. The program activities should be listed separately for each user category, e.g. General, Player, Administrator. The activities that are exactly the same for Player and Administrator will then go under the "General" heading. After the learner has lists of all the activities, each activity must then be broken down into a series of events. Then the learners should break down each event into component parts.

Task

A User interface for a language game is given where the user must match the columns by using drag and drop. The learners must list all the events involved in playing this game in sequence.

Notes and Teaching suggestions

The learners must understand that they must work out the details of every aspect of the program before starting to create files or write the program code. The better the design, the less work there will be during Phase 3. Let the learners concentrate on the events by asking themselves: "What must happen when the user chooses an activity?" Note that there are lots of events involved when a database is used like deleting records, changing records, adding records, etc. All must be listed. Explain the difference between event driven programming and sequential programming. This will help the learners to understand that each event can have different actions. Encourage the learners to place all events in sequence to make the planning and later coding easier. We suggest that the learners do the task individually.

Lesson 4: Phase 2 - Algorithms I

In this lesson learners learn to work out the tasks the program must perform during each event, and then write the step by step instructions, called the algorithm, for each task. These instructions must be in sequence and can be written in pseudocode. Although pseudocode is less formal than programming language, it does use control structures for sequence, selection and iteration. Each step in the algorithm is an instruction that describes input, processing, storing or output of data. The word "Input" is used for user input from the keyboard, and "Input selected" is used for input from the mouse. "Display" is used for output on the monitor. If a database is used as permanent storage, we use "store" and "fetch" to get some info from it. "Read" and "Write" is used for a text file.

Task

A game of noughts and crosses is given as the scenario. The learners must do the algorithm for 1) the repetition, 2) the IF statement for one specific move, and 3) the IF statement for the next given move. After the examples, the task can be done individually by each learner or a group can do one part of the task and assessment can be done in class context.

Notes and Teaching suggestions

Make sure that the learners use indentation in their algorithms when using selection and iteration structures. This will make it easy to see which instructions belong inside the loop and which is outside during coding. Algorithms are a very important part of programming and we suggest that a few simple examples are done in class to practice before doing the task. The learners must be able to determine all actions that are repeated and put them inside the loop and where there is a condition to be met, the action goes to a selection structure.

Lesson 5: Phase 2 - Algorithms II

The learners learn how to show processing instructions from their algorithm. They identify modules that can be re-used and write the algorithms for those modules and to then call the needed module. The learners also learn to ask themselves the right questions which will help them to determine what must be done with each instruction. This lesson also teaches the learners to draw flowcharts to illustrate the flow of all the events.

Task

A part of an algorithm is given to the learners. Questions 1 and 2 just need to be answered from the given algorithm while question 3 is a short function to be coded.

Notes and Teaching suggestions

It may be a good idea to revise some hardware essentials here. The RAM only keeps the data temporary until it is used by the CPU or stored permanently on the hard disk. The CPU can only compare data, do calculations or move data as instructed by the program. The better the learner instructs the CPU, the better the program will be. Selection structures are used to compare data. Revise the different operators that can be used to test a condition. Also revise the use of variables during calculations. Let the learners get used to using the arrow (←) to indicate the results of a calculation being returned to a variable. Some learners get confused with the data types when doing calculations. This is ideal place to teach them to determine the data type for each result and to check if the variable was assigned to the correct data type. Parameter passing as input or output within modules can be revised here very effectively. In the beginning learners may have difficulty with flowcharts, so we advice the teacher to do a few in the class. If the learners really understands the flowcharts and become comfortable with it, it can make their coding much easier. We suggest that questions 1 and 2 of the task are answered and discussed in class. Question 3 can be done individually or in their groups.

Lesson 6: Phase 2 - Drawing up IPO Tables

It is expected from the learners to draw up an IPO table from an algorithm. A few Interfaces are provided during this lesson where they take the learner through it step by step to determine which actions are input, where processing has taken place and where one could expect output. The learners are also guided through error messages in the case of a wrong input.

Task

In the given scenario the teacher will enter an exam mark out of 75. The program will have to convert it to 100 and then give a symbol as output. The learners must draw up the IPO table for this scenario.

Notes and Teaching suggestions

When using a MaskEdit component, the learners must understand that the actual word is stored and not the symbols that are displayed in the MaskEdit. The learners must use "**Available**" if the data input is not by the user but from another resource like a database or even a constant. An IPO table must be done for EVERY algorithm. Learners must give each table a name and summarise what is happening in the event. We recommend that the learners use a word processing program for their algorithms and for the IPO table. It saves time to copy and paste some information to the IPO table. Learners may now find it necessary to alter some algorithms.

Lesson 7: Phase 2 – Storing Data in IPO Tables

This lesson teaches the learners to identify primitive, structured and abstract data types to be used for temporary storage in RAM. The learners get a short revision of all the data types. Learners must also indicate where they will validate data, check for errors and handle exceptions.

Task

An IPO table is given to the learners for a player or an administrator logging in to a program. Only a player will choose a grade from the combobox. The learners must add the following to this IPO table:

1. The available temporary storage structures
2. Any files necessary for permanent storage
3. Explain how selecting a grade can assist with data validation

Notes and Teaching suggestions

Revision of the different data types is necessary here. Primitive data types like Integers, Real, String and Boolean are used for variables. Arrays and Records are structured data types which contain a number of elements. Abstract data types are classes used to create an object. Learners must understand why a password must rather be stored in an object and not in a text file. The IPO for an object should be done separately. Let the learners first identify all temporary storage and then the permanent storage. Data validation and error checking must always be done on user input. Exception handling can occur anywhere in the program and learners are not always confident with this term. A good example is to prevent the program from crashing if the database is already open. Get the learners to be familiar with the "Try ... except"-code. When the learners learn to add the input- and output parameters in the functions' and procedures' IPO table, they will understand parameter passing better. We suggest that this task should be done section by section with feedback before going to the next section. Some parts of this video can even be replayed to ensure understanding.

Lesson 8: Phase 2 - Classes, Objects and Data Flow I

In this lesson the learners learn how to draw up class diagrams for objects that will be used in a program. They learn how to use RAM more effectively. This lesson teaches the learners how data fields in a record can be accessed directly as well as how data fields in an object can be accessed. They will understand that data encapsulation isolates and protects the data from the rest of the program. Sometimes it is necessary that a class is a subclass of another class in which case it inherits all the fields and methods of that class with some additional fields and methods of its own. The learners distinguish whether a member of a class should be private or public.

Task

A text file with learner names and test marks is given. From this information, the learners have to:

1. Create an object class names clsTMarks
2. Create a class named TMarks in this unit with prescribed private fields
3. Create four public methods of which two are constructors and two are functions

Notes and Teaching suggestions

All learners have experienced that a program gets slow or hangs due to too much data in the RAM. They need to understand that by using classes they use the RAM more effectively as it keeps the data in just one location in the RAM. We recommend that the teacher revises certain aspects of classes like attributes and methods during this lesson. A few examples in class can help them to understand that the members of a class can be private, protected or public. We suggest that there is feedback on the first part of the task before the learners go on to part two.

Lesson 9: Phase 2 - Classes, Objects and Data Flow II

This lesson is all about creating class diagrams and simple data flow diagrams. From the IPO tables we can determine what fields and methods are needed. First the learners have to decide which fields will be private if any and declare them as usual. Next the methods have to be declared with their parameters in brackets. Constructors are used with parameters. Once every object is described in a class diagram, the learners must create a flow diagram which must indicate where the data will come from and go to, and where the data will be stored.

Task

The learners must draw up a class diagram for a player object which must store the player's username and two totals, one for the number of questions and one for the correct answers. The totals must increase after each game. The username and the score as a percentage must be returned.

Notes and Teaching suggestions

Make the learners aware to destroy and object when the program closes to release it from the RAM. Highlight that the "f" in front of a field helps to distinguish the field from variables that will be used in the methods. Why we use constructors and their characteristics are important to discuss here, especially that a constructor always clears the storage fields in a new object by default. Because a constructor always clears the fields by default all fields start with a value of 0, so fields in a constructor only have to be initialised if the learner wants it to start with a different value. Encourage the learners to use colour in their diagrams to show interaction between the level 0 and level 1 diagram. We recommend that the task is done is sections with feedback before continuing to the next section.

Lesson 10: Phase 2 - Database Design I

Before the learners can create a database, it has to be designed first. In this lesson they learn to determine which data must be stored permanently in a database or text file by referring to all the documentation of the previous lessons. When designing the database, each separate data item is a field. A table name must be given to all the fields that belong together.

Task

Each record in the text file given is a game played by a specific player. The fields are the date of the game, the level, the score and the terms that the player got wrong. The learners have to answer six questions on ways to find the name of the text file in which the program has to update the statistics of that player.

Notes and Teaching suggestions

Each field in the database must have a data type. If learners are not sure to which table a field belongs, let them still write it down as "Other". This will make sure that they will not forget about it. Although text files can be useful, there is no way of validating data, avoiding errors in input, identifying specific data items, or creating relationships between data items. Encourage the learners to add functions that the client didn't ask for, but will add value to their program.

Lesson 11: Phase 2 - Database Design – final requirements

In this lesson about database design, the learners learn about sorting data items, primary keys, lookup tables and queries. Learners are also reminded not to store facts not to be used or which can be obtained somewhere else. If an attribute seems to belong to two entities, learners must decide which entity owns that attribute. After the different tables have been created, decide on a primary key for each table. Only then the learners can draw up ERM diagrams to show the structure of their database.

Task

The learners must design a database table for only the GamesPlayed table. This table of information must include the fields, data types, maximum data size, validation, input masks and lookup fields. The relationships between the tables are given.

Notes and Teaching suggestions

It is important that learners are quite confident with primary and foreign keys as well as relations between tables. Revise these concepts before or during this lesson. For a computer each entity must have one unique attribute. Explain why it is better for the name and surname to be in two different fields. Let the learners get used to the guidelines given in the lesson to draw up ERM diagrams. The foreign key field can reference only one primary key in another table, but the primary key can be referenced by the foreign fields of many records. One-to-one relationships don't need foreign keys. This task should be done individually by each learner.

Lesson 12: Phase 2 - User Interface Design

The learners have to identify all the screens of their program and also indicate how the program will move from one screen to another. This lesson teaches them to do it by using a flowchart with a diagram or they can use a presentation program like PowerPoint. They learn to use different shapes for different parts of the screen. The learners can do one screen for each event or put more than one event on a screen. Each screen must be given a self-explanatory name. Then each screen must be explained fully in a table.

Task

The GUI design for a Multiple Choice test is given. The learners draw up a table to explain the components to be used, which data can be accessed from this interface, the user actions required as well as the validation requirements.

Notes and Teaching suggestions

Let the learners determine the start and the end of each screen and indicate it in their flowchart. Learners must not indicate any components but illustrate it as options. During the task the learners must keep in mind that the client has to approve the screens. Much emphasis should be on user-friendliness.

Teaching Guidelines – interactive PC lessons

Lesson 1: Phase 3 - Database Tables

In this lesson the learners are taught to build a multiple-table database. By means of demonstrations with the Access program, the learners learn step-by-step to analyse data to be stored in order to construct multiple tables in the 3rd Normal Form (3NF). They also learn to set the appropriate data type as well as properties for each field. The learners have to answer a quiz after different topics to ensure that they understood what has been done.

Task

The scenario for the additional task is a SA Chess Championship Tournament which requires a 3NF database. Each player will enter his/her own result after each game. The information that is required is described. The learners must name the field and tables.

Notes and Teaching suggestions

We suggest that the slower learners take notes also while they do these lessons. Encourage the learners to read everything on each screen as important hints are usually also given. There are altogether 7 quizzes which are essential for learners to do. Learners must understand why they have a question wrong. These lessons will really help them to understand Normalisation of a database if done properly. Additional there is a worksheet as well as a task which can be done after the completion of this lesson. The assessment of the worksheet and the task can be used for formal or informal assessment. The task has a suggested answer. There may be small variations in the learner answers as long as all the rules learned in the lesson have been followed.

Lesson 2: Phase 3 - Database Relationships

The construction of the database table started in lesson 1 gets completed in this lesson. The learners learn to create relationships between the tables by creating foreign keys and to identify data which does not belong in the database. The learners do one-to-one as well as one-to-many relationships. Referential and data integrity are maintained by setting parameters.

Task

The learners are given six tables. The name of each table is at the top outside each table. The primary key is indicated with the underlined field. The tables are about shops selling books to customers from their different branches over the country. The learners must draw lines to indicate the relationships between the different tables as well as the type of relationship.

Notes and Teaching suggestions

We suggest that the demonstrations in this lesson be done practically by each learner. Each learner can replay the demonstration until they can do the required without having to look back to the lesson material. The practical task must not be done on the computer, but with

pen and paper. The emphasis is here on the relationship itself and not on the method of doing it with the computer.

Lesson 3: Phase 3 - GUI Components

This lesson teaches the learners to use the outside-in approach while designing their main screens. They are working from the overall design in towards the detail. The learners are introduced to good GUI design principles keeping in mind the structure, simplicity, visibility, feedback, tolerance and re-use of coding.

Task

The learners have to list at least 10 good GUI design principles.

Notes and Teaching suggestions

This lesson gives a lot of information which seems to be quite obvious and afterwards appear not to be. We suggest that the learners take notes while they go through this lesson. It can also be a good idea if the learners can have this lesson on their computers to refer back if necessary. The task is very essential in this lesson and it is recommended that feedback is given in class so that learners can add to their list. This list of design principles should be at hand while they are working on phase 3. This list might also come in handy for the teacher during the assessment of phase 3.

Lesson 4: Phase 3 - The GUI and the User

In this lesson the learners learn how to present their information on a screen. They also learn about multimedia learning principles. The different ways to allow a user to enter data are dealt with in this lesson as well as the advantages and disadvantages of each. The learners also get useful hints on which to provide guidance to a user.

Task

The learners have to name and discuss each of the principles for the presentation of information briefly.

Notes and Teaching suggestions

This lesson teaches a lot of useful information which may make it worthwhile if the learners take notes for later references while looking at this lesson. To put this lesson into practice, we suggest that the learners take their PAT of Grade 11 and discuss or evaluate the screens by means of the principles in this lesson. The teacher can also select any programs (maybe one with a good GUI design and one with bad GUI design) and discuss them in class. The worksheet and task can be done individually and used as an informal assessment.

PAT phases 1 & 2

Lesson 1: Investigation and Analysis of a problem

In this lesson we begin to work out how to create software to solve a client's problem, that existing software can't solve. We start with some preliminary research to find out what our client needs and then decide what kind of software we should create. As we investigate and analyse 'the problem', we'll take a look at the need for custom software, as opposed to off-the-shelf software. And we'll demonstrate the steps involved in developing custom software.

Lesson Outcomes:

By the end of this lesson, you should be able to:

- Recognise the need for custom software
- Understand the steps involved in custom software development
- Outline the key aspects
- Explore and evaluate educational sub-genres
- Identify end-user expectations for this educational software genre

Curriculum Links:

L O 4: Programming and Software Development

Lesson Notes

Before you start designing the program for your PAT, you need to do some initial research. As you investigate and analyse a problem of a client related to information technology, you will:

1. Meet with the client to find out:
   - Who the client is
   - Who the user or end-user of the software will be
   - What the problem of the client is
   - What the client is using at the moment
   - What the requirements of the software are

2. Create a Problem Statement.

3. Investigate the problem fully, to determine
   - If there is any other suitable software available, like freeware or shareware
   - What the shortcomings of the existing software are
   - If there is any Open Source software that can just be modified to solve the problem.
   - If you have to create new software
   - Whether you have to create new software using existing software, add-ons or plug-ins.
   - What the key areas of the existing software are and whether it meet the client's needs
   - What the nature of the existing software is
   - What the limitations of the current software are
   - What challenges the existing software has

Open Source Software is software operated under a license that permits users to study, change, improve and sometimes even distribute your version of the software.

Add-ons and **plug-ins** are software that add functionality to a program, but cannot be run independently.

Some useful tools to use during the investigation include:

- Survey – set some questions to ask to several possible end-users.
- Internet – look at similar programs.
- Questionnaires – set questions on paper for possible end-users to answer.
- Interviews

Record-keeping is very important. Keep all questions, answers, personal particulars, search words, the URLs of websites, print-outs of web pages, dates of activities, etc.

Task

Thabo has gone to see the IT teacher to find out if she has any software needs. The teacher explains that her learners are not getting good marks in her IT terminology tests. She wants to provide a program that can help them find all the terms in one place and learn them easily.

1. Suggest a suitable problem statement
2. Write down at least 2 questions for each area Thabo must investigate to get to the nature of the problem
3. List current efforts to solve the problem (without the use of software)
4. List the limitations and challenges in the current solution.

Lesson 2: Identifying the Requirements

In this lesson we will identify the essential requirements of the software needed to solve the client's problem. First you look at the notes you made while you gathered information during the investigation to find the client's need. To show the client what you have in mind you have to write down the exact requirements for input, processing, storage and output.

Lesson Outcomes:

By the end of this lesson, you should be able to:

- ✓ Identify requirements for:
  - User input from mouse or keyboard
  - Processing like calculations, comparisons, searching and sorting etc.
  - Storage in files on hard drive
  - Output by the program to the monitor
- ✓ Describe a possible solution

Curriculum Links

L O 4: Programming and Software Development

Lesson Notes

It is important to discuss all the requirements in detail to make sure that you know exactly what to do in your PAT as well as to show the client what you have in mind. The requirements involve:

1. **Input** is any information given by:
   - Typing (using a keyboard)
   - Performing an action like selecting an option (using a mouse)
   - Indicating a process must start or stop (using a keyboard or mouse)
2. **Processing** involves:
   - Calculations
   - Logical comparisons with a result of True or False
   - Moving data to a storage location
3. **Output** - Only state what will be seen on the monitor or heard by the user. You don't have to mention the type of output components in Phase 1.

Note: Be careful not to think item by item. Rather see each requirement in context. Divide your program into different stages (one screen at a time) and list all the requirements (steps) for that screen. If you are not quite sure how it will work, make general suggestions. Also give all the different steps for each option on a screen and state whether it is Input, Processing, Storage or Output.

Keep in mind that your PAT must include:

- A database with multiple tables
- One or more text files
- Objects and classes

Describe a possible solution to the problem which will include:

- A summary of the requirements
- One or two paragraphs

- Scope of the program (what the program **can** do)
- Limitations of the program (what the program **can't** do)

Remember to attach all the research you used in the investigation as well as a list of references!

Task
The netball coach wants a program that will help users to record all the games in the netball season, check that all the rules have been followed and calculate the results of each game. Write a solution for her, showing all the requirements for input, processing, storage and output.

Lesson 3: Program Specifications

In this lesson we will determine what your program must do according to the already set requirements in phase 1. You will learn how to list the activities the user will be involved in as events without worrying about detail.

Lesson Outcomes:

By the end of this lesson, you should be able to:

- Explain the difference between program requirements and program specifications
- List expected events for a given program scenario
- Classify events according to the sequence they occur in
- Break down general statements of events into component parts

Curriculum Links

L O 4: Programming and Software Development

Lesson Notes

The **program requirements** (phase 1) are what your client wants the user to do with the program, while the **program specifications** are when you use those requirements to figure out what the program **must** do.

1. List the events.

To list the events you ask yourself what activities will the users be involved in when they use the program. If you have different categories of users, you must list their events separately, and the events that all users are involved in under a "General" heading. Everything that is necessary to make your program work becomes a separate event. There should be a list of events for each option on each screen. To list the events, you should ask:

- What else needs to happen (what should the user be able to do)
- What is essential for the program to work if the user clicks on e.g. a button (list everything that can happen in the program)

Also remember the events of your database! If you make use of queries, list them as events too. The same goes for reports.

An event driven program is when the user can choose any part of the program. We can jump around different events in one program. To start an event in a program, the user must **select** an **option**. Thus, an event is triggered by some action.

2. Break down the events in component parts.

Make sure that you list every event of your program in sequence to make planning easier.

The Practical Assessment Task

www.internetpolyglot.com

Task

Here is the User Interface of a language game that asks you to match columns. You must match a word with its correct translation by using drag and drop. When you are done you click the continue button to check the answers, and see your score.

Name the events involved in playing this game, in sequence.

Lesson 4: Algorithms I

In this lesson we are going to work out what tasks the program must perform during each event, and write down the instructions for each of these tasks.

Lesson Outcomes:

By the end of this lesson, you should be able to:

Break an event down into its component steps, clearly showing:

- sequence
- selection
- iteration

Curriculum Links

L O 4: Programming and Software Development

Lesson Notes

An algorithm is a step by step solution to a problem, and can be written in pseudocode. Pseudocode is writing commands and instructions using English words and programming language strucures. It is easy to translate an algorithm into any program language.

Program Code	Pseudocode
Very formal	Less formal
Strict syntax rules	Sentence structure
Uses program control structures – sequence, selection, iteration	Uses program control structures – sequence, selection, iteration

Each step in the algorithm is an instruction that describes what the program must do during an event. It describes input, processing, storage and output. The format of an instruction is: DO WHAT (WHERE)?

Step 1: Break each event into tasks and write an instruction for each task.

SEQUENCE - To show sequence, we write the instructionsin the order in which they must be done.

SELECTION - The program has to select one set of instructions to follow if a given condition is true, and a different set of instructions to follow if the condition is false.

```
If (condition true)
THEN
(set of instructions)
ELSE
```

REPETITION – A set of instructions which must be repeated.

```
REPEAT (a certain number of times)
    Instructions
REPEAT UNTIL (a given condition is true)
    Instructions
```

The Practical Assessment Task

Step 2: Describe for each instruction whether it is input, output, processing or storage.

Input: The data must be moved from the keyboard into the RAM by the user.
Input selected: The data must be moved from the mouse into the RAM by the user.
Display: The data must be moved from the RAM to the monitor.
Print: The data must be moved from the RAM to the printer.
Play: The output must go to the speakers.
Record: Input must come from the microphone.
Fetch: Data moves from the database to RAM. (database input)
Store: Data moves from the RAM to the database
Read: To move the data from a text file into RAM. (input from textfile)
Write: To move the data from RAM to a textfile.

0	X	b
b	0	
Next 0		X

Task

A game of noughts-and-crosses is played by two players. One person places a nought in any square, then the other places a cross. The first person to get three of their symbols in a line (row, column or diagonal) wins the game.

1. List the instructions for the game, showing sequence and repetition only.
2. Show the selection or IF statement that was used for the move where X placed a symbol to try to prevent 0 from winning, in the bottom right swuare. Remember to cater for a normal move.
3. The next move is in the coloured filled block. X is going to lose, as he can only block one of the two swuares, shown by a small "**b**". Adapt your IF statement for this condition, making sure it caters for all other conditions.

Lesson 5: Algorithms II

In this lesson we are going to learn how to show processing instructions. We will finish our algorithm by writing the processing instructions. During this process we will revise modular coding like functions and procedures and then draw a flowchart to illustrate the flow of events.

Lesson Outcomes:

By the end of this lesson, you should be able to:

- Explain modularity, identify re-usable modules of code and write algorithms for the modules
- Draw a flowchart to illustrate a flow of events

Curriculum Links

L O 4: Programming and Software Development

Lesson Notes

Writing processing instructions:

A CPU can:

- Perform logical operations on data (**Compare** – return a value of true or false) Comparing data to test a condition using an IF statement

>, >=, <, <=, =, <>, AND, OR, NOT	OR	Plain English

- Perform mathematical operations on data (**Calculate** – return a result, stored in a variable). The variable named on the left will store the result of the calculation on the right.
- Send and receive data (**Move**)

.

Ask the following questions during input in your algorithm:

- Must I **compare** or check this input?
- Must I do any **calculations** on this input to get other values from it?
- Can I discard this input after I have compared and calculated, or must I **move** the input to a storage location?

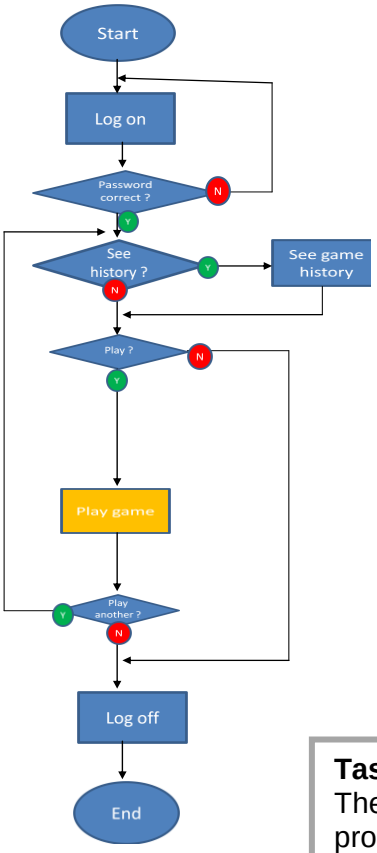
A **module** (procedure or function) is an independent unit that can be used to construct a more complex structure. Add a module of code to your program by writing an instruction to **call** the module. Code modules may require input from a program or may give output to the program that calls them which is known as **parameter passing**. Output can be from a procedure or ***Return*** a single output from a function.

Procedure	**Function**
Performs a certain task and can give output	Performs a certain task and always returns exactly one output
Can change things in program, if given a reference to the things to be changed	Does not change things in your program
	Returns a value which you can use to change things in the program.

The Practical Assessment Task

Hint: If one algorithm is getting very long, look for parts where several instructions are used to do one task. That task can be removed from the algorithm and made into a module. Each algorithm can just call for the task when it is needed.
Use a word processor to write your algorithms.
When you write your program, complete the code for your modules before your write any other code.

Now draw a flowchart to illustrate the flow of events: sequence, selection and iteration.



Task
The following is part of an algorithm for a program event:

GameSore = 6 (Integer)
TotalScore = **FetchTotal**(username)

TotalScore = **NewTotal**(GameScore, TotalScore)
GameScore, TotalScore (data type Integer);
username (data type string)

1. Are FetchTotal and NewTotal procedures or functions? Give a reason for your answer.
2. What is the data type of the value returned by FetchTotal? Explain.
3. Write the re-usable module for NewTotal, which simply adds the score, and returns the new total. Show all data types.

Lesson 6: Drawing up IPO tables

In this lesson we are going to learn how to use our algorithms to draw up an IPO table, including storage, for each event. This will also help us to discover the most suitable components for input and output in the GUI.

Lesson Outcomes:

By the end of this lesson, you should be able to draw up an IPO table for a given event, to include:

- Input
- Processing
- Output

Curriculum Links

L O 4: Programming and Software Development

Lesson Notes

Input components are: text boxes, radio buttons, check boxes, combo boxes and buttons.
Output components are:

- Text panes – for large amounts of information
- Labels – for small amounts of information
- Message dialogues – to draw the user's attention

The program takes the input from the components, **stores** it in temporary memory (RAM), **processes** it, sometimes using other available data and **saves** it in permanent storage (hard drive). Then the program gives feedback or **output** to the user.

Use the word "Available" for additional data which does not have to be input by the user.
Available means:

- values stored in **constants**
- values stored in **non-local variables** or **arrays**, by the processing which occurred in another event
- the data in the **fields** (attributes) of an object in RAM, which can only be accessed by calling the **method** of that object
- the data in **files** like databases and text files stored on the hard drive

Follow the following steps to do the IPO tables:

1. Do an IPO table for every algorithm.
2. Give each table a heading and insert a comment just under it to summarise what is happening in the event.
3. Cut and paste your algorithm into the processing column.
4. Decide what data is needed and whether it must be input by the user, or be available to the event. Then fill in the input column.
5. Complete the output column and show what the user must see after the processing has been done.
6. Refer to components, objects, variables, constants, arrays and files that **store available** data during input and output.

Task

Your teacher needs a program to allow her to record marks quickly and accurately. The exam is worth 75 points. She will enter an exam mark and needs to know the letter grade (A+, A, B, C, etc.) for the exam. Draw up an IPO table to show the solution to this teacher's problem.

Lesson 7: Storing Data in IPO Tables

In this lesson we will identify the different data types used for storage. First we will revise what primitive, structured and abstract data types are. We will identify the different data types for each event of every algorithm and then indicate it in our IPO table. Then we will identify file types and structures to be used for permanent storage. The last step necessary to complete our IPO tables is to indicate where and how you will validate data, check for errors and handle exceptions.

Lesson Outcomes:

By the end of this lesson, you should be able to:

- Identify primitive, structured and abstract data types to use for temporary storage
- Identify file types and structures to use for permanent storage
- Indicate where data validation, error checking and exception handling are required

Curriculum Links

L O 4: Programming and Software Development

Lesson Notes

1. Identify each of the following, for temporary storage, and add them to the IPO table:
    - **Primitive data types** like integers, real, string and Boolean must be assigned to all **variables**
    - **Structured data types** which can be **arrays** (an indexed list of data items of the same data type) or **Records** (a complex data structure consisting of a number of fields or elements, each of which can be a different data type)
    - **Abstract data types** are the **classes** we use to create **objects** that are pre-defined by the class. They consist of storage fields and methods of behaviour.

    *Hint: Do not write data types in the row where the processing algorithm is described. Write them in the input and output columns or add the variable to the **Available** row and fill in the data type there.*
2. Identify the data you want to store permanently. In a separate document use the headings “Text files” and “Database” and indicate the name of the file/table and everyting that will be stored in there.
3. Insert a row at the bottom of each IPO table and indicate each of the following:
    - **Data validation**
    - **Error checking**
    - **Exception handling**

Task

Study this IPO table (next page), which contains the algorithm for logging in to a program and indicates user input and output. In this scenario, the person logging in can either be a player or the administrator. The administrator would not select a grade, but would leave the default value of “choose a grade” in the combobox.

1. The available temporary storage structures have not been listed in the “Available” section. Study the processing column to determine available **temporary** storage structures, then write them down.
2. Name any files you will need for permanent storage
3. Explain how selecting a grade assists with data validation

The Practical Assessment Task

INPUT	PROCESSING	OUTPUT
	Available	

grade
username
attempt
	Display grades
	
	Input grade, username, attempt
	IF (grade ='Choose a grade') AND (username = 'Admin') THEN
	IF attempt = AdminPassword THEN
	IsAdmin = true
	CanProceed = true
	Display message
	ELSE
	Display message
	Exit event
	ELSE
	REPEAT for all records in learner table
	IF (username = stored username) THEN
	IF (attempt = stored password) THEN
	CanProceed = true
	FirstName = stored first name
	CurrentGrade = grade
	Display message
	ELSE
	Display message
	Exit event
	ELSE
	IF NOT((username = stored username) OR
	(username = 'Admin')) THEN
	Display message
	Exit event
		Choose a grade
		Grade 10
		Grade 11
		Grade 12
		
		Welcome Admin
		
		Incorrect password
		
		Welcome FirstName
		
		Incorrect password
		
		Not on the system

Lesson 8: Classes, Objects and Data Flow I

In this lesson we explore the objects and methods of classes in object-orientated programming. We will distinguish between records and objects and then draw up class diagrams for objects that will be used in a program. We will learn a better way of using the RAM by creating one or two objects that can each hold many data items.

Lesson Outcomes:

By the end of this lesson, you should be able to:

- Understand classes and objects used for data storage in object orientated programming
- Explain the advantages of data encapsulation
- Differentiate between private and public members of a class

Curriculum Links

L O 4: Programming and Software Development

Lesson Notes

Use the RAM more efficiently by:

- Store separate data items for an event in one logical place
- Protect the data from unwanted modification or deletion
- Pre-define tasks that collect, modify, use or delete the data
- Free the RAM easily when the data is no longer needed

Using **objects** for data storage is called an **object-orientated program.**
An object-orientated program takes data from a database, text file or input data and stores it in objects in RAM.

Data fields in each of the following can be accessed directly by:

Record	**Object**
Using normal programming code	Can be accessed by program code, indirectly
Creating SQL text to be used by an ADO query component in RAM connected to the database	An instruction in a program can call a pre-defined method of the object
Using database management software like MS-Access	Cannot decide what can or can't be done with the fields
	Methods are functions and procedures, or modules of code pre-defined what may be done with the data
	No other access to the data is possible

This concept of isolating and protecting the data from the rest of the program is called **data encapsulation**.

Step 1: Define the classes used to create the storage objects you mentioned in your IPO tables.

A class has:

- **Attributes** (fields or properties) to store data (use variables as the fields)
- **Methods** (or behaviours) to give instructions for ways of handling the data (procedures, functions, constructors)

A class can be a subclass of another class, which means it can inherent all the fields and methods of that class. It has some additional fields and methods of its own.
Members of a class can be:

- **Private** – only methods of the same class can access them.
- **Public** – any code can access them.

If your program has to store or retrieve data in an object of its class, it will have to pass that data as parameters and will only be able to do that if the public methods it calls allows it to happen.

Dlamini#62#84#90#
Du Toit#22#60#96#
Jacobson#49#78#50#
Mahlangu#48#29#32#
Moeketsi#98#60#95#
Molebane#83#24#9#
Mthethwa#6#76#18#
Naidoo#80#85#57#
Ndebele#47#2#15#
Niels#14#60#61#
Reyneke#83#26#47#
Robertson#0#99#86#
Sharp#83#75#67#
Van de Vyfer #88#2#71#
Williamson#93#96#72#

Task

You are creating a program using object-orientated programming to display and make calculations with test marks. The data is saved in a file called **Marks.txt**

1. Create an object class (another unit) named clsTMarks.
2. Define a class named Tmarks in this unit. This class must contain the following private fields:
   Surname : String;
   Test1 : Integer;
   Test2 : Integer;
   Test3 : Integer;
3. Create the following public methods:
   Constructor.Create.overload;
   Constructor.Create(pSurname : String; pTest1, pTest2, pTest3 : Integer) : overload;
   Function getAverage : String;
   Function showMarks : String;

Lesson 9: Classes, Objects and Data Flow II

In this lesson we are going to inspect the IPO tables to determine what fields are needed for each class, and what methods we should have to draw up a class diagram. We will learn how to create and place an object in RAM when the program opens and destroy it from RAM when the program closes. Finally, we will use a flow diagram to show the movement or flow of data identified in all the IPO tables.

Lesson Outcomes:

By the end of this lesson, you should be able to:

- Create class diagrams
- Create simple data flow diagrams

Curriculum Links

L O 4: Programming and Software Development

Lesson Notes

Constructors:

- Are defined like a procedure
- Can only create an object of that class type, and set initial values of fields
- Are called like a function
- Can only return a reference to the object created, stored in a variable of class type

A constructor always clears the storage fields in a new object by default. That means, all fields start with a value of zero for number data types, empry or null for string types, or false for Boolean types. So, you only initialise fields in a constructor if you want them to have a particular value.

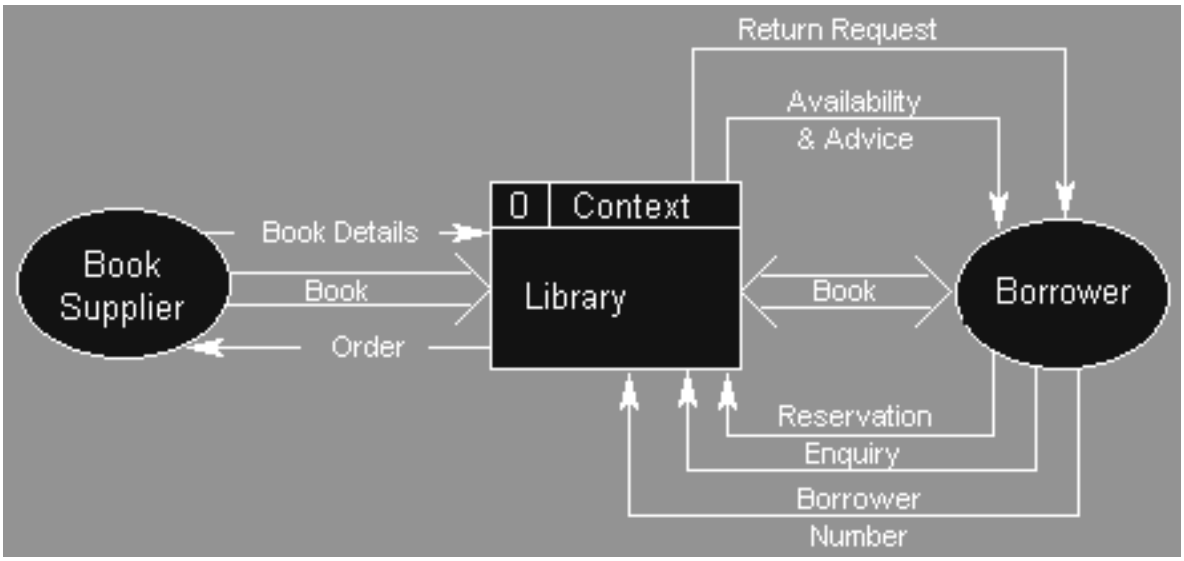
Components of a **Data Flow Diagram**:

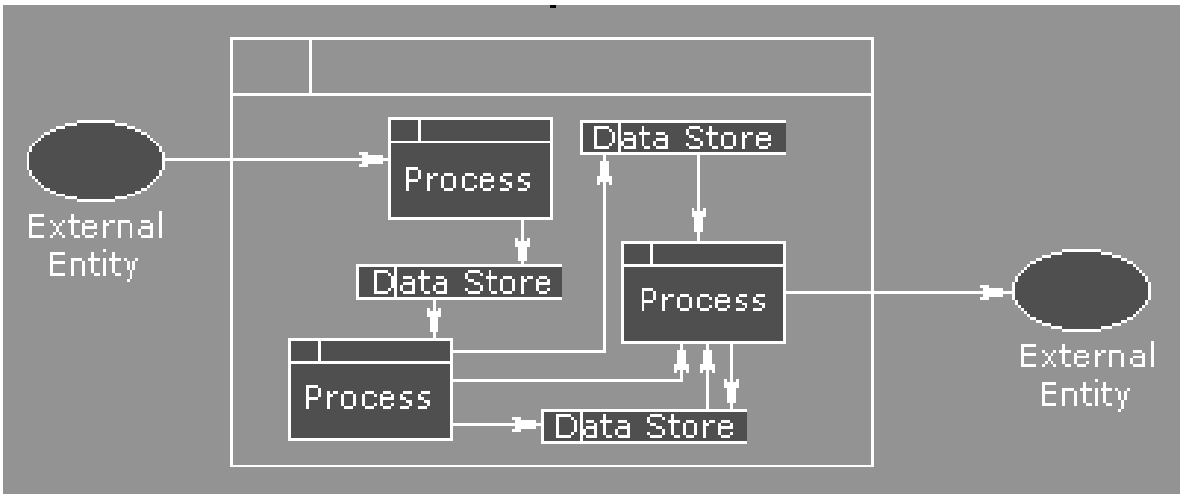
- **Entities** – outside the system boundaries, give input (source) or receive output (destination)
- **Processes** – events that occur in the system, receive input and deliver output
- **Data flows** – depicted by one-way or two-way arrows, can be labeled to name the data items being transferred
- **Data storage locations** – temporary structures in RAM and permanent files on the hard drive

Steps to create a class diagram (from the top) and a data flow diagram:

1. Write a heading with the name of the class
2. State the Private fields (if any) and their data types. An “f” in front will help to distinguish the fields from variables that will be used in the coding.
3. List the Private methods (if any)
4. List the Public methods. Each method having a name and a type – either procedure or function. The bracket after the method name show what input is required. Empty brackets show there is no input, but every method must have brackets. Each bracket is followed by a colon (“:”) and then the data type of the output, if any.
5. Draw up a Context Diagram to show the system as a single process with the external entities. It does not show storage.
6. Draw up a Level 0 Diagram that shows general processes (main events) and data storage (like database, text file, object) inside the system. Processes can send data to each other, e.g. one process may call another to check the validity of data.
7. Draw a Level 1 Diagram to illustrate any of the more complicated processes and data stores from level 0 – ignore the simple ones.

The Practical Assessment Task





Task

Draw up a class diagram for a player object, which must store the player's username and two totals. One total is for the number of questions attempted and the other is for the number of answers which were correct. These totals must be increased after each game has been played. The object must be able to return the username for messages, and a number that will be shown to the player as the percentage of questions correctly answered.

Lesson 10: Database Design

In this lesson we are going to discover which types of data will go to text files and which to a database. Then we will start with the design of a database. First, we will use the notes in the separate document for databases, which you saved during designing your IPO table as well as the class diagrams and data flow diagrams.

Lesson Outcomes:

By the end of this lesson, you should be able to:

- Identify data that must be stored permanently
- Determine the data that must be stored in text files
- Indicate how data structure will be shown in each text file

Curriculum Links

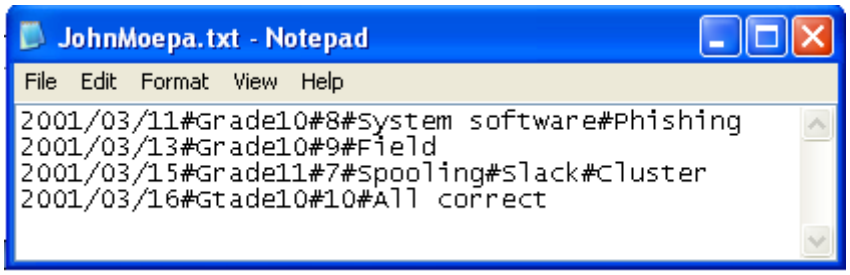
L O 4: Programming and Software Development

Lesson Notes

There is no way to protect data in a text file. So, you can't store sensitive personal data in a text file. Also, all the data from a text file must be accessed or transferred in sequence, line by line. In some cases it is best to store the data in two places – a database to record the data and a text file for exporting data to another program for processing.

To design a database, look at your IPO tables as well as your class diagrams and data flow diagrams. Identify all the data that must be permanently stored. For the database use the following steps: Write down each separate data item as a field

Write the names for the fields you will use - Show the data type for each field - Give a table name for items that you are certain must be stored together



Use the text file about a game to answer the task questions. Each line is the record of a game played by a player. The fields are the date on which the game was played, the level at which it was played, the score, and the terms that the player got wrong. At the end of each game, the program will find the correct text file for that player, and add it as their latest data, as a new record. The program will have to find the correct file for each player.

The Practical Assessment Task

Task

Say whether each of the following cases would be a good way to get the name of the text file, and give a reason for each answer.

1. Ask the player to type in the file name after each game.
2. Use the **file open dialogue**, and allow the player to select the correct file.
3. Read the first name and surname from the database table, and construct a string for the file name after each game.
4. Store the file name as a string in the player's record in the database.
5. Store the file name in the player's record, and transfer it to a Player object when the player logs on.
6. This data could be manipulated, if it is first read into an array. Give an example of what use you might have for part of each line, for examples the grade and score, or the terms which were answered incorrectlv.

Lesson 11: Database Design

In this lesson we will look at the final requirements for our database. We will learn about sorting data items (fields) into the correct table, setting a primary keys and foreign keys for each table. We will also learn how to draw up an EMR diagram for a database.

Lesson Outcomes:

By the end of this lesson, you should be able to:

- Sort data items according to the entity of each item
- Determine natural fields and add suitable primary keys
- Add foreign fields where relationships between tables must be created
- Determine appropriate data types for fields
- Prepare a list of queries, reports and table manipulation

Curriculum Links

L O 4: Programming and Software Development

Lesson Notes

Golden rules for storing data in a database:

- Don't store any fact that are not going to be used
- Don't store any fact that can be obtained from other stored facts
- Don't store any fact in two different places
- Break down a complex fact into a few simple, useful facts

Step 1: Put the fields into the correct tables.

Step 2: Determine all the essential natural fields. Add fields you may have forgotten and remove fields you don't need. Split fields, e.g. the name and the surname.

Step 3: Add a suitable Primary Key to each table. For a computer to distinguish between different entities of the same kind, each entity must have one unique attribute. If there is no natural one, add an attribute that uses simple sequential numbers.

Step 4: Show the relationships between the tables, using an ERM diagram.

Step 5: Create another table to indicate how field properties should be set like data type, field size, validation rule, input mask and lookup tables

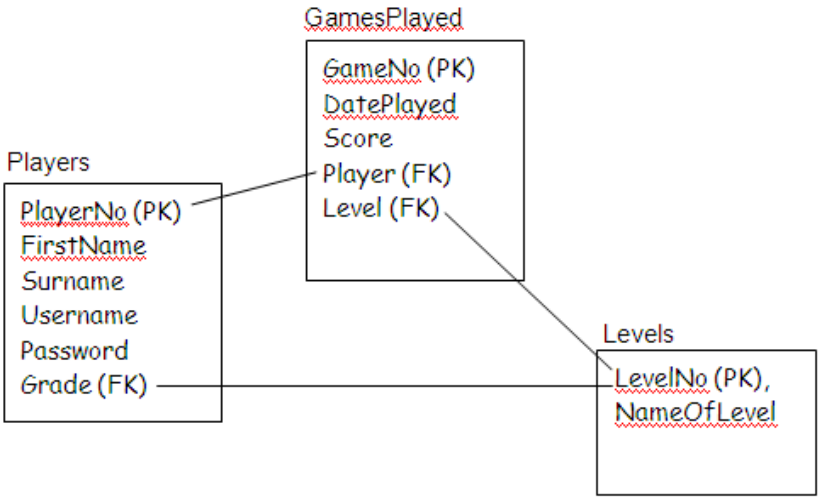
Hint: *Think of attributes as **nouns** and relationships as **verbs**.*

If you find a fact you need to store that seems to belong to two entities, you must decide which entity really **owns** that attribute.

Basic guidelines for an ERM diagram:

- Entity sets (tables) are drawn as rectangles
- Draw diamonds for a relationship. Entity sets participating in a relationship are connected to the diamond with a line.
- Attributes are drawn as ovals.
- Underline the Primary Keys. Remember the Foreighn Key field can reference only one Primary Key in another table.

The Practical Assessment Task



FIELD	TYPE	SIZE	VALIDATION	INPUT MASK	LOOKUP
GameNo (PK)	autonumber				
DatePlayed					
Score					
Player(FK)					
Level (FK)					
GameType					

Task

Draw up a table of information for the GamesPlayed table. In your table, indicate: Field name, data type of the field, field size or format, Validation rules to be set, Input Mask you might need and whether data must be looked up from one of the other tables, or from a list you will type in. Notice that there is an extra field GameType for recording whether the game played was a multiple choice or a match the column game.

Lesson 12: User Interface (GUI)

In this lesson we will design all intended screens for our program. First we will draw up a flowchart to identify all the screens of our program and show how the program will move from one screen to the other. Then we will use a table for each screen to indicate all the components that will be used.

Lesson Outcomes:

By the end of this lesson, you should be able to:

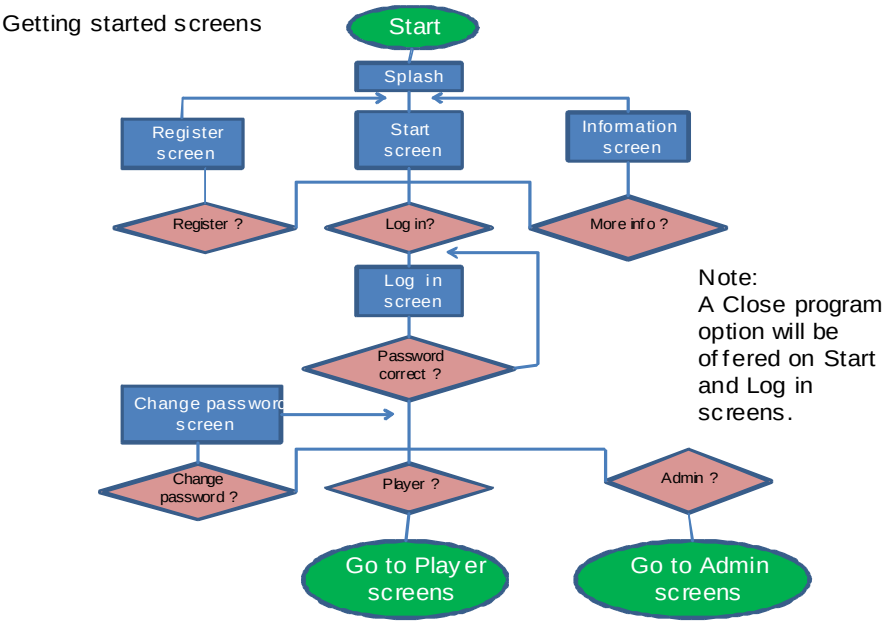
- Illustrate and explain the navigation between screens
- Indicate the accessibility of different screens to different categories of users
- Illustrate and explain the intended screens fully

Curriculum Links

L O 4: Programming and Software Development

Lesson Notes

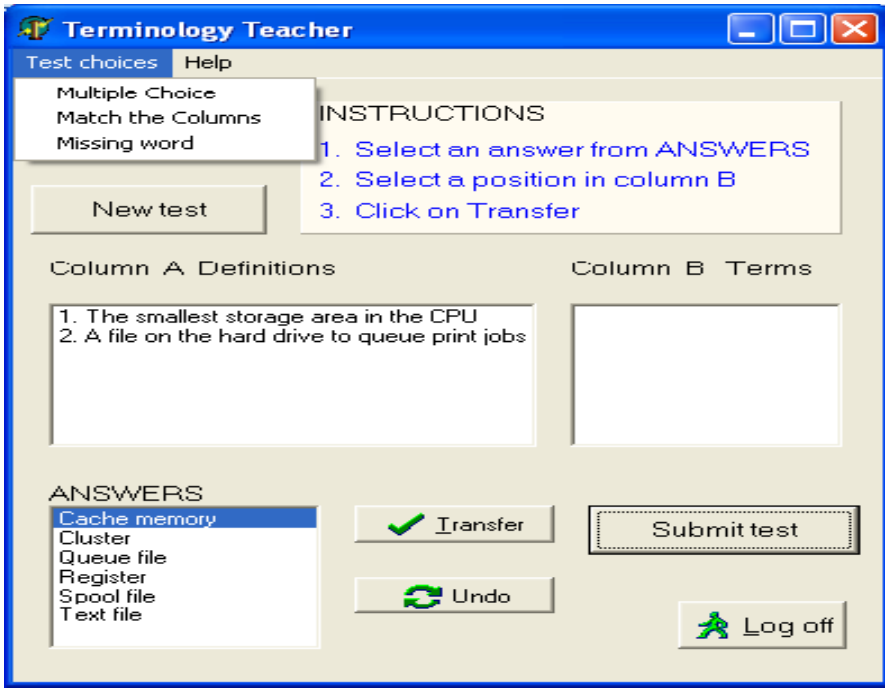
Use diagramming software of a presentation program like MS PowerPoint to draw the flowchart and diagrams. Create all your shapes first and then drag them into place. In the flowchart you need to show the **Start** and the **End** of the flow from screen to screen, using **oval** shapes. The actual screens are **rectangles** and the user's choices or actions for navigation are shown as **diamonds**. Add **lines** to show the actual flow with **arrows** for the direction of the flow. Don't show components on the flowchart.



Illustrate the design or layout for each screen on the flowcharts with a table.
Use the following heading to describe each screen:

- **Description** – Explain what the screen will be used for.
- **Components** – Mention all the components you intend to use and for what.
- **Data** – Say which stored data will be accessed from this screen
- **Actions** – Which user actions are offered within the screen and for navigation to other screens
- **Access** – List all the users that will have access to this screen
- **Validation** – Specify the components that must be validated if any/

When you decide on components for the screen, remember that they must store user input and output. They must also give information to help the user perform actions or they must offer the choices of action for the user.



Components	
Data	
Actions	
Validation	

Task
This is the design for a screen for a Multiple Choice test.
Draw up a table to explain the components to be used, the data that must be accessed from this screen, the user actions, and the validation requirements.

The Practical Assessment Task

PAT phase 3

LESSON 1 WORKSHEET
DATABASE TABLES

The principal of your school is allocating different teachers to different projects like the matric farewell, prize giving, etc. Put each of the following fields under the correct heading in the given table: Allocating teachers to project; DateOfEvent; Project; ProjectName; Teacher

Lesson Outcomes

- Analyse data to be stored in order to construct multiple tables
- Set appropriate data types for each field in each table and set field properties
- Construct lists of lookup values to input data

Event	Entity	Natural Attribute

Which of the following data types will be the most suitable for each field?

FIELD	DATA TYPE
ID	
Description	
DateOfEvent	
Fee	
LogoPicture	

Which field property will be used in each case:

DESCRIPTION	FIELD PROPERTY
A spesific input format is required	
This field cannot be left blank	
An initial value given although user can change it	
The input is limited between certain boundaries	

Task

1. The organising committee of the SA Chess Championships requires a database where each player can record the results of their games on a computer as soon as each game is finished. Design a multiple-table database where all the tables are in the Third Normal Form (3NF). Give appropriate fieldnames for each description. Indicate the primary key for each table (it may be necessary to add fields).

Remember that the two players will each enter the results separately. This can then also be a cross-reference of the results. The following information will be needed:

- The name and the surname of the player who enters the result
- The name and the surname of the opponent
- The player who enters the result will have his/her own username and password
- The date that the game was played
- The union (province) that each of the players belong to
- In which round was this game played
- The section or age group in which the players compete
- As a win is indicated as a 1, only the score of the players entering the score is necessary
- The colour (white or black) of the player whose score is entered
- The boys and the girls for each age group are separated – bear in mind that a girl may compete in the boys section

PAT Phase 3: LESSON 2 WORKSHEET
DATABASE RELATIONSHIPS

Lesson Outcomes

- Identify areas where one table in a database needs to reference the primary key field of another table
- Identify data that must be stored outside of the database
- Determine the nature of all relationships
- Use database tools to create foreign fields
- Establish links between tables in with one-to-many relationships tables
- Use database tools to import data
- Establish links between tables in with one-to-one relationships
- Set parameters in the relationships to maintain referential and data integrity.

What is the function of the foreign key in a table?_____

_____

_____

_____

What type of relationship will there be between the IT teacher and the Grade 12 IT students?

_____

_____

_____

How can you ensure that a relationship will stay valid when changes are made to some records?

_____

_____

_____

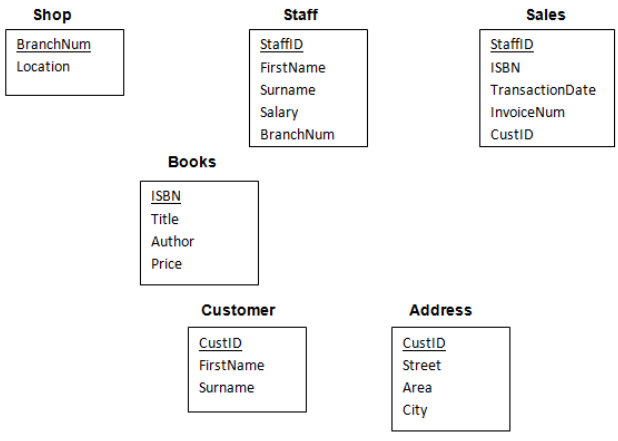
Which takes up the most storage space, a database or a text file?

_____

TASK

The following are database tables created for a bookshop with several branches ountrywide.

1. Indicate the relationships between the different tables.
2. Also indicate whether each relationship will be one-to-one (1 – 1) or one-to-many (1 - ∞).



The Practical Assessment Task

PAT Phase 3: LESSON 3 WORKSHEET
GUI COMPONENTS

Explain what the outside-in approach means.

_____

_____

_____

List three components that the programmer uses to store the data temporarily in RAM.

_____

_____

_____

Lesson Outcomes

- Select appropriate main container components (main screens) to hold groups of other components
- Create the necessary links to navigate between the container components
- Select appropriate components to store data temporarily during program execution
- Change the default settings of these components to optimise their functionality in a given context

List three components that the user can use to manipulate some of the data stored in RAM.

_____

_____

TASK

1. List at least 10 good GUI design principles.

The Practical Assessment Task

PAT Phase 3: LESSON 4 WORKSHEET
THE GUI AND THE USER

Lesson Outcomes

- Select appropriate secondary container components to group related components.
- Select appropriate components to input data from both keyboard and mouse.
- Change the default settings of these components to optimise their functionality
- Validate input data where necessary to prevent errors in data manipulation
- Select appropriate components to give information to the user, to guide users to manipulate the data as intended by the program.

The multimedia principles is applicable on one of the following (circle or tick):

A. Sound; Video
B. Images, Animation; Text
C. All Media

State whether each of the following principles is applicable on A, B, or C:

Learning is easier when information in different media is all seen at the same time, rather than one by one.	
Don't make the user read and listen to the same information.	
Learning is easier if you present only material to be learned and not unnecessary added information.	
If we see and hear about the same thing at the same time it has a stronger learning effect as the brain can use both modes together.	
Learning is easier if the information is grouped together for one topic.	

What is the purpose of a pop-up menu?

Give two disadvantages of a Radio Group.

Name one advantage of a Combo Box.

TASK

1. The presentation of information principles must be applied to each screen, form or window throughout the program, keeping the intended user in mind. Name and describe each principle briefly

PAT phases 1 & 2

LESSON 1 TASK ANSWERS

The following is a 'model' answer and shows all the aspects that should be included:

Problem Statement

Many learners find it difficult to learn IT subject terminology, and don't perform well in the multiple choice and match the columns questions in an exam. There does not seem to be any software which could help to solve the problem. I am going to design software to be used by IT teachers and learners. The purpose of the software will be for the user to see the correct definition of a term frequently, so that over time the terms are learned. The software will present IT terms and definitions for grades 10 to 12 as random questions in a test. The advantage of the software will be that it will assist teachers who don't have time to keep giving and checking homework, and it will help IT learners to find the terminology from all three grades in one place. The learners will learn by drill-and-practice, but in a fun way. The user will be able to select the correct term or definition, depending on which type of test they choose, and will get immediate feedback after the test to see how well they have done, and what the correct answers were. I will have to make sure that I have collected and stored all possible terms, and that the teacher can edit my terms and definitions, or add new terms. I will also make sure the teacher can check the learners' progress.

Background

Learners are not doing well in tests involving multiple choice questions and matching of the columns which involve IT terminology. They find it difficult to learn the terms in the limited time available in class, and if they were absent when some terms were taught, they don't know them. Some learners don't understand more difficult terms, and some find they don't really know which terms to learn.

The teacher gives lists of terms to learn for homework, makes learners draw up lists of terms for each chapter in a textbook, and gives learners lists of terms to look up on the internet, but many learners still don't seem to learn enough.

Learners don't have sufficient time, and they have only their current grade's textbook to refer to and so can't learn work from a previous grade. Many different definitions are given for the same term so they are not sure which one is correct. Some of the terms they have to learn are not in the text book.

A comprehensive list is needed for each grade, to ensure that all terms are defined in clear, unambiguous and understandable language, and to make the information quickly and easily accessible to a learner, preferable stored in one place.

Key aspects of the problem

The curriculum document will be consulted to ascertain which terms must be known. All terms for grade 10, 11 and 12 must be collected, with clear unambiguous definitions for each. Terms which are not listed in the document will be added by the teacher if necessary.

The program will be a drill-and-practice type, so that learners can learn by repetition. The program will offer multiple choice tests, where the definition is provided and the term must be chosen, and match the columns, where the correct definition must be found to fit the term. It is very important that the user should see the correct terms and definitions matched, so these will be shown after each test. While the program will not be a game, an effort will be made to add some entertainment value, so that the learners enjoy the experience.

The program will be used by learners in grades 10 to 12, ages 16 to 18, male and female, who are taking IT as subject, and who are familiar with using computer software. Learners indicated that they want to see their scores and to see which terms they got wrong.
The program will also be used by the teacher to check learners' progress and to add, edit or delete terms. Therefore, the program must also keep a record of all learners who take the test.

LESSON 2 TASK ANSWERS
The following is a 'model' answer and shows all the aspects that should be included:

Log in
Register
The user will input the Username, Password (I)
The program will store the new password (S)
A message will display on the screen that welcomes the new user (O)
Log on
The user will input the Username, Password (I)
The program will check if the password match (P)
The program will "Welcome" the user (O)
Change Password
User enters Username, Password (I)
The program will check if the password matches that of Administrator (P)
Administrator enters username to change (I)
Administrator enters new password (I)
The program will store the new password (S)
Message that the password has changed will be displayed on the screen (O)

Game Info
The user enters the date of the game, the teams (I)
The user enters the players of both teams (I)
If a player scored, it is entered to the player's name by the user (I)
The program will update the Total Score after each score entry (P)
The program displays the updated score on the screen (O)
If a player has a penalty against her, it is entered to the player's name by the user (I)
All the entries are stored by the program (S)

Game Results
The user selects the date of the game (I)
The user chooses the required results like Game Score, Total Penalties, Total Score for Players, Average Goals (I)
The computer will calculate the results (P)
The program will save all the results (S)
The results will be displayed on the screen or printed (O)

LESSON 3 TASK ANSWERS
Start the game
Drag the correct definition to the term
Get your results
End the game or start another

LESSON 4 TASK ANSWERS
1. REPEAT UNTIL
(three 0s in a line) OR (three Xs in a line) OR (all squares filled)

Place 0 in an empty square
Place X in an empty square

2. IF (two opponent's symbol are in one line)
   THEN
      Place own symbol in line
   ELSE
      Place X in any empty square

3. IF (two opponent's symbols in one line) AND (two opponent's symbols in another line)
   THEN
      Give up (or Exit the game)
   ELSE
      IF (two opponent's symbols are in one line)
         THEN
            Place own symbol in line
         ELSE
            Place X in any empty square

LESSON 5 TASK ANSWERS

1. Functions – they will return only one value to this calling event.
2. Integer – the returned value will be placed in the variable TotalScore, which can only hold integers.
3. Function NewTotal(Score, Total : Integer) : Integer *// Input:Score, Total; Return:Integer*
   NewTotal = Total + Score
   Return NewTotal

LESSON 6 TASK ANSWERS

INPUT	PROCESSING	OUTPUT
Mark	If Mark > 89	A+
	If Mark > 79	A
	If Mark > 69	B
	If Mark > 59	C
	If Mark > 49	D
	If Mark > 39	E
	If Mark > 29	F
	If Mark < 30	G

LESSON 7 TASK ANSWERS

The following is a "model" answer and shows all the aspects that should be included:

1. Available
   Grade, Admin in combo box (strings)
   Usernames in records in Learner table in **database**
   Array of strings to store existing usernames

2. Files Needed
   Text File: AdminPass.txt
   Database: Learners table (FirstName, CurrentGrade)

3. Selecting a grade from a combo box ensures that the user does not give the wrong input for example make a spelling mistake.

LESSON 8 TASK ANSWERS

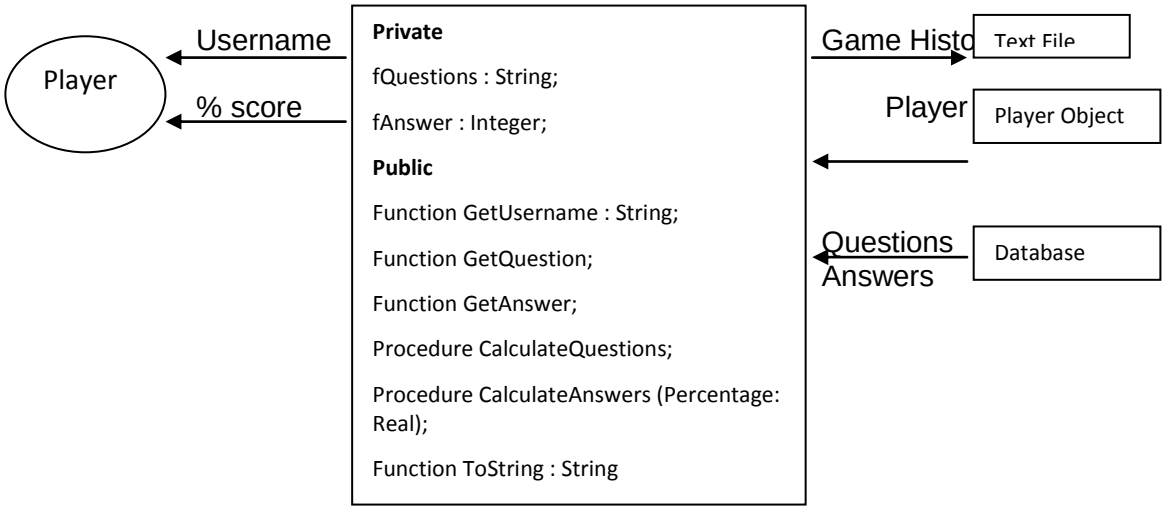
1. Unit clsTMarks

2. **Class: TMarks**
   fSurname : String;
   fTest1 : Integer;
   fTest2 : Integer;
   fTest3 : Integer;

3. **Public**
   Constructor Create.overload;
   Constructor Create(pSurname : String; pTest1, pTest2, pTest3 : Integer): Overload;
   Function getAverage : String;
   Function showMarks : String;

LESSON 9 TASK ANSWERS

The following is a "model" answer and shows all the aspects that should be included:



LESSON 10 TASK ANSWERS

1. Not good – The player would not want to constantly retype the filename, and could also make a mistake.
2. Not good – A player could select someone else's file if he gets a bad score, and add the poor performance record to that file.
3. Not good. This is pointless repetition. The file name should only be created once.
4. Good – As long as the file name is read into a non-local variable when the player logs on. The database should not have to be queried each time. Hard drive access is slow.
5. Good – The Player object can store the two names for other uses as well (e.g. message), and use a method to create the filename. Reading the name from RAM will be faster than reading it from the database file on the hard drive each time.
6. The grades and scores could be used to find the player's average score on each level. Wrong answers from all games the player has played could be read into another array, (IF the answer is NOT All Correct). Duplicates could be removed, and a list of terms could be screen) or as a printout.

LESSON 11 TASK ANSWERS

Table: GamesPlayed					
FIELD	**TYPE**	**SIZE**	**VALIDATION**	**INPUT MASK**	**LOOKUP**
GameNo (PK)	Autonumber				
DatePlayed	Date/Time		Yes – not Null	dd/mm/yy	
Score	Number				
Player(FK)	Text	20	Yes – not Null		
Level (FK)	Text	10			Grade
Game Type	Lookup	25			Type of game

LESSON 12 TASK ANSWERS

Components	Menus to choose the type of test; Panel to display the instructions; RichEdit/Listbox to show the questions; ComboBox where the user can choose an answer; RichEdit to display the users choice of answer; Labels will indicate each RichEdit; Buttons for user actions
Data	Questions and answers from database table
Actions	New Test must clear all RichEdits and ComboBox; Transfer will display the user's choice in the RichEdit; Undo will reverse the last action; Submit Test will compare answers with the database; Log off will close the program
Validation	If the user Submit Test without answers for all the questions, a message will ask them to do all the questions before submitting

PAT phase 3

LESSON 1 WORKSHEET ANSWERS

Event	Entity	Natural Attribute
Allocating teachers to project	Project	DateOfEvent
	Teacher	ProjectName

FIELD	DATA TYPE
ID	Text
Description	Memo
DateOfEvent	Date/Time
Fee	Number
LogoPicture	OLE Object

DESCRIPTION	FIELD PROPERTY
A spesific input format is required	Input Mask
This field cannot be left blank	Required
An initial value given although user can change it	Default Value
The input is limited between certain boundaries	Validation Rule

TASK ANSWERS

1. Different tables with fields (underlined indicates primary key):

GamesPlayed
GameID
Date
Round
Score
Colour

Player
PlayerID
FirstName
Surname
Username
Password
Union

Opponent
OpponentID
FirstName
Surname
Union

Section
SectionID
Section
GenSec

LESSON 2 WORKSHEET ANSWERS

What is the function of the foreign key in a table? The foreign key in a table allows one table to look up values in a related table.

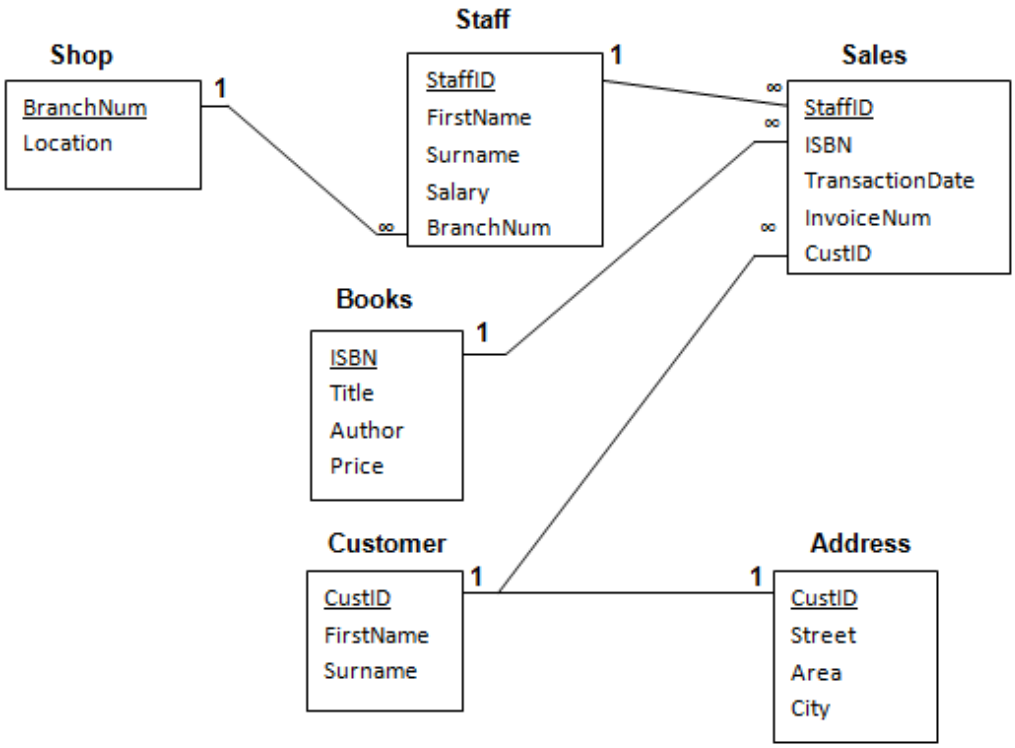
What type of relationship will there be between the IT teacher and the Grade 12 IT students? A **one-to many** relationship

How can you ensure that a relationship will stay valid when changes are made to some records? Use the **Enforce Referential Integrity**

Which takes up the most storage space, a database or a text file? A **database** takes up the most storage space

TASK ANSWERS

1 & 2: (next page)



LESSON 3 WORKSHEET ANSWERS

Explain what the outside-in approach means. The outside-in approach means that you first design all the main screens and then do the detail later.

List three components that the programmer uses to store the data temporarily in RAM.
Structures like variables, arrays, objects, non-visual components (*Any three*)

List three components that the user can use to manipulate some of the data stored in RAM.
Any GUI components like button, radio button, checkbox, editbox, menu, etc.

TASK ANSWERS

1. List at least 10 good GUI design principles.
   Organise parts of the program in a logical way. Group related things together. Make similar things resemble one another. Tell the user what to do in clear, simple and unambiguous language. Provide easy to follow shortcuts and links. Make all options visible and understandable at once. Don't give too much to be read at a glance. Give the user feedback as messages. Keep the user informed about what is happening. If errors have been made, explain how they can be corrected. Allow the user to cancel, undo or redo things. Provide code to prevent errors where possible. Allow the user to correct errors. Allow for all reasonable actions. Re-use components and behaviours where possible. Maintain consistency throughout the program.

LESSON 4 WORKSHEET ANSWERS

Learning is easier when information in different media is all seen at the same time, rather than one by one.	B
Don't make the user read and listen to the same information.	A
Learning is easier if you present only material to be learned and not unnecessary added information.	C
If we see and hear about the same thing at the same time it has a stronger learning effect as the brain can use both modes together.	A
Learning is easier if the information is grouped together for one topic.	B

What is the purpose of a pop-up menu? (Any one of the following) It gives brief tips when help is needed. It appears where it is needed (in context)

Give two disadvantages of a Radio Group. The Radio Group uses lots of screen space. The Radio Group shows unnecessary information after the choice has been made

Name one advantage of a Combo Box.(*Any one of the following*). The Combo Box takes up little space on the screen. The selection of the user is visible after it has been made. There are no display of unnecessary information after the choice has been made.

TASK ANSWERS

1. The presentation of information principles must be applied to each screen, form or window throughout the program, keeping the intended user in mind. Name and describe each principle briefly.
   Clarity – The information is correct and is quickly and easily understood.
   Conciseness – The information includes only essentials, not "waffle" or "padding".
   Discriminability – The information is not "lost" in the decorative elements.
   Legibility – The information is visually readable with e.g. a clear font, a big enough font size and clear images.
   Consistency – The same type of information is presented in the same way in different places or at different times.
   Detectability – The user is directed towards information immediately, and doesn't have to look around the screen to find it.

The Social and Ethical Impact of ICTs

Series overview

This series discusses the impact of Information and communication technologies (ICTs) on various areas of our lives. New technology improves communication by new ways to communicate and replacing older bulkier hardware with smaller, more efficient hardware. ICTs make it possible to access an endless amount of information 24 hours a day, seven days a week. But with this volume of knowledge available to us, and in some cases about us, we need to know how to distinguish between valid and authentic information and useless data and how to protect ourselves against various forms of malicious software, or "malware" as well as cyber-crimes like bandwidth theft, information theft, Hacking, Social engineering, Identity theft, Phishing, hoaxes and DOS attacks. Various physical and psychological problems that can occur when using ICTs incorrectly are mentioned. The series also discusses the impact of ICTs on business, politics, jobs, culture and the environment.

Curriculum links

Learning outcome 1 covers the development of computer hardware and software to improve performance of computer systems. In this series the focus is on Learning outcome 2 and 3 where the social and ethical implications of Information and Communication Technologies, and e communication is covered.

The Assessment Standards of Learning Outcome 2 require the learners to:

- critically analyse information retrieved from the Internet
- critically assess Internet security and its implications
- propose and justify strategies to protect the value of information produced and communicated by an organisation
- state and discuss how the latest technologies facilitate human interaction

The Assessment Standards of Learning Outcome 3 require the learners to:

- analyse the effects of the use of computers across a range of application areas
- research and report on the ethical issues relating to the use of computers
- comment on the use of computers in providing solutions to issues of national and international importance
- comment critically on the social, political, economic and other consequences of search engines and group communications
- explain the responsible use, purpose and significance of any new computer developments

Educational approach

Many teachers neglect this section because of time constraints and learners under estimate the section since it sounds like common sense when you read the content. You are however encouraged to show these videos to your learners and follow it up with group discussions using the tasks as a starting point.

Learners who do not have access to the latest technology will see and hear various experts explain how new technology is used and how it affects various aspects of our lives. The videos include for example interviews with a doctor who explains physical problems that can occur when you use computers incorrectly, a graphic designer who shows how he uses computers in his designs and even a professional hacker!

The lesson Notes give a summary of the main concepts in the videos, but should be used together with the video material.

The Social and Ethical Impact of ICTs

Series at a glance

Lesson title	Lesson Outcomes By the end of this lesson the learner should be able to:
1. **The Impact of information and communication Technologies on our lives**	• Explain the general advantage of new computer developments • Discuss examples of the effect of computer developments on communication • Discuss examples of the effect of computer developments in areas of our daily lives
2. **How Technology can Benefit or Harm the User, the Environment and Society**	• Evaluate how the availability of information can benefit society or create problems for us. • Discuss the economic, political and cultural implications of technology for society. • Deduce physical, psychological and sociological implications of technology for society and • list ways to minimize the impact of technology on the environment
3. **Cyber Threats and Safeguards to Protect you against Attacks**	• Distinguish between cyber threats like Spyware, key loggers, phishing, identity theft, social engineering, hacking, Internet-related fraud scams, hoaxes and DoS attacks. • Describe the purpose of safeguards like backup, Firewalls, Passwords, user access rights and privileges, digital signatures, digital certificates, encryption, SSL and Public Key encryption

Teaching Guidelines

Lesson 1: The Impact of information and Communication Technologies on our Lives

This lesson discusses the impact of Information and communication technologies (ICTs) on various areas of our lives. The video shows examples of how new technology improves communication by allowing e-mail, sms, mms, chat, blog, social networking and the sharing of documents in the cloud. You will see insets of older bulkier hardware versus the new technology like smartphones, tablets and new ways of communication like RSS, blogging, podcasts, sending tweets, Skype and video conferencing, will be illustrated. Interviews with an expert in education, a TV expert, a graphics expert, a music expert and the owner of a small business give insight into the impact of ICTs on these fields. You will see examples of how ICTs impact on travel when using, for example GPS, RFID tags and computer chips in cars.

Before you show the video, ask the learners how old the oldest computer that they have ever seen is. Since the learners were only born in the nineties, you can perhaps share with them what your first computer looked like. Then discuss the most modern computers and phones that they have seen or used and follow up with different ways the learners use to communicate. If necessary ask whether they use RSS, blogging, podcasts, Twitter, Skype, Mxit etc.

Task

The task requires learners to give advantages and disadvantages of several aspects of ICTs, which are becoming increasingly important and widely-used in everyday life. Not all of these applications are covered in the lesson, so learners need to conduct some reading and research, and apply the learning of this lesson.
This task assesses Learning outcome 2 assessment standard 9 and Learning outcome 3 assessment standard 2 and 11. The task is open ended. The possible answers in the Lesson notes are a guideline but any other meaningful advantages and disadvantages are acceptable

Lesson 2: How Technology can Benefit or Harm the User, the Environment and Society

ICTs make it possible to access an endless amount of information 24 hours a day, seven days a week. But with this volume of knowledge available to us, and in some cases about us, how do we keep ourselves safe from people who will use information against us and sort out which bits to believe and which bits to write off as pure fiction? This lesson discusses the authenticity and validity of information, warns against unethical behaviour like plagiarism and piracy and explains why you should not give out your personal details on the Internet. Various physical and psychological problems that can occur when using ICTs incorrectly are mentioned. The video also discusses the impact of ICTs on business, politics, jobs, culture and the environment.

It is important to make the lesson interactive so that the learners can make the information their own. Before you show the video, you can read a hoax e-mail and ask them whether they think it is true to introduce the concepts of authenticity and validity. You can also ask them how many hours per day they spend in front of the computer, on Facebook or playing games so that they will identify with the physical and psychological problems discussed in the video.

Task

The task also expects learners to extend their learning from the lesson to common environmental needs. Encourage them to make their own deductions after carrying out further reading and research. This task focuses on Learning Outcome 3 assessment standards 6 and 11, which cover ethical issues and the impact of technology on the environment.

The possible answers in the Lesson Notes are a guideline but any other meaningful advantages and disadvantages are acceptable

The task can be extended by asking the learners to determine the authenticity and validity of a given e-mail. Examples can be found at http://www.hoax-slayer.com/

Lesson 3: Cyber Threats and Safeguards to Protect you against Attacks

This lesson discusses various forms of malicious software, or "malware" like Trojans, Spyware and Key loggers as well as cyber-crimes like bandwidth theft, information theft, Hacking, Social engineering, Identity theft, Phishing, hoaxes and DOS attacks. The video even included an interview with a professional hacker! The last part of the lesson explains how you can protect your data by using up to date anti-virus and anti-spyware software, not to allow people you do not trust to use your computer, not to use public computers for your Internet banking or online shopping, making Backups, using firewalls and strong passwords.. You can also make use of digital signatures and digital certificates as well as data encryption to protect confidential information.

The Social and Ethical Impact of ICTs

Task

The task enforces aspects of user safety and preventive approaches against threats and attacks. The task focuses on Learning Outcome 2 assessment standard 6, about internet security. The possible answers in the Lesson Notes are a guideline but any other meaningful answers are acceptable. Make sure learners can distinguish between different cyber threats and that they use the proper terminology when discussing them.

We apologise for the spelling error at the start of each video lesson. Please note that the series is called The Social and Ethical Impact of ICTs.

(ITCs is incorrect)

The Social and Ethical Impact of ICTs

Lesson 1 The Impact of Information and Communication Technologies in our Lives

This lesson discusses the impact of Information and communication technologies (ICTs) on various areas of our lives. The video shows examples of how new technology improves communication by allowing e-mail, sms, mms, chat, blog, social networking and the sharing of documents in the cloud. You will see insets of older bulkier hardware versus the new technology like smartphones, tablets and new ways of communication like RSS, blogging, podcasts, sending tweets, Skype and video conferencing, will be illustrated.

Interviews with an expert in education, a TV expert, a graphics expert, a music expert and the owner of a small business gives insight into the impact of ICTs on these fields. You will see examples of how ICTs impact on travel when using for example GPS, RFID tags and computer chips in cars.

Lesson Outcomes

By the end of this lesson, learners should be able troubleshoot:

- Explain the general advantage of new computer developments
- Discuss examples of the effect of computer developments on communication
- Discuss examples of the effect of computer developments in areas of our daily lives

Curriculum Links

Learning Outcome 2 and 3

Assessment standards:

12.2.09	States and discusses how the latest technologies facilitate human interaction.
12.3.02	Analyses the effects of the use of computers across a range of application areas.
12.3.11	Explains the responsible use, purpose and significance of any new computer developments.

Lesson notes

Information and communication technology (ITCs) are used everywhere, for example in homes, offices, schools, hospitals, cars, buses, trains, planes, sports stadiums, libraries, television studios, laboratories and coffee shops to store, find, send and receive information.

When you compare earlier technologies with those which used today you find that the first commercial computers were extremely large and expensive, they were also not very user friendly, so the average person would have struggled to operate them. Today devices like laptops, cell phones, ipads and mp3 players deliver greater speed, increased processing power, more user friendly interfaces, greater mobility, reduced size, reduced costs, greater efficiency and increased variety. The video shows pictures of various old and new devices – see how many you can identify.

The advances in technology have over the past decade radically changed how we communicate in our daily lives. Where once we would have hand written and posted a letter at the nearest post box, we now use cell phones, sms, mms, email, Mxit, BBM, Twitter, blogs, wikis, podcasts, RSS feeds, YouTube, Skype, video conferencing and Social Networking sites like Facebook. Being able to communicate beyond the confines of our immediate vicinity has allowed us to grow, become more aware of others, more tolerant, more aware of our common humanity, more likely to work together for the benefit of the

The Social and Ethical Impact of ICTs

species as a whole. Since news travel fast, we have the chance to respond and influence events quickly and effectively.

ICTs have made a large impact on Education where schools use computerised systems for the administration of the school, interactive boards, e-books and electronically supported learning and teaching where content is delivered via the Internet, intranet/extranet, audio or video tape, satellite TV, MP3 Players, Palm pilots, CDs and DVDs, like Mindset does. Students can e-mail teachers with questions or submit assignments online. Universities and some schools have portals where students can access their time tables, exam results and notes online.

ICTs are also used in travelling, for example you can research various destinations to plan a trip or read blogs of people who visited the destinations before. You can book hotels and tours online and even use GPS to find your way around in foreign cities. Technology is also used to control traffic lights, toll bridges and parking pay stations and cars and aeroplanes make use of technology to function.

The video will show interviews with a small business owner, specialist in entertainment industry, TV expert, music industry expert to explain how ICTs impact on their environments. Remember: with power comes responsibility. As users of these technological instruments we need to be sure that we evaluate and critically assess all the information we receive and send out into the world.

Task

Give 2 advantages and 2 disadvantages for each of these:

a. a virtual school where you do not go to school but rather download all lessons from the internet and submit assignments online
b. e-banking
c. shopping online

The Social and Ethical Impact of ICTs

Lesson 2 How Technology can Benefit or Harm the User, the Environment and Society

ICTs make it possible to access an endless amount of information 24 hours a day, seven days a week. But with this volume of knowledge available to us, and in some cases about us, how do we keep ourselves safe from people who will use information against us and sort out which bits to believe and which bits to write off as pure fiction?

This lesson discusses the authenticity and validity of information, warns against unethical behaviour like plagiarism and piracy and explain why you should not give out your personal details on the Internet. Various physical and psychological problems that can occur when using ICTs incorrectly are mentioned. The video also discusses the impact of ICTs on business, politics, jobs, culture and the environment.

Lesson Outcomes

By the end of this lesson, learners should be able troubleshoot:

- Evaluate how the availability of information can benefit society or create problems for us.
- Discuss the economic, political and cultural implications of technology for society.
- Deduce physical, psychological and sociological implications of technology for society and
- List ways to minimise the impact of technology on the environment

Curriculum Links

Learning Outcome 2 and 3

Assessment standards:

12.2.05	critically analyses information retrieved from the Internet.
12.2.06	critically assesses Internet security and its implications.
12.2.08	proposes and justifies strategies to protect the value of information produced and communicated by an organisation.
12.3.02	analyses the effects of the use of computers across a range of application areas.
12.3.06	researches and reports on the ethical issues relating to the use of computers.
12.3.09	comments on the use of computers in providing solutions to issues of national and international importance.
12.3.10	comments critically on the social, political, economic and other consequences of search engines and group communications.
12.3.11	explains the responsible use, purpose and significance of any new computer developments.

Lesson notes

ICTs make it possible to access an endless amount of information 24 hours a day, seven days a week. Blogs, social networking sites, Wiki's and Websites have little or no regulation or censorship. This means that people can publish whatever content they like. But with this volume of knowledge available to us, and in some cases about us, how do we keep ourselves safe from people who will use information against us and sort out which bits to believe and which bits to write off as pure fiction?

To determine whether information is valid, ask yourself whether it is

- relevant
- current
- related to the current topic

- presented within context
- fact based
- un- biased
- can be cross referenced

To determine whether information is authentic, ask yourself:

- Can you believe the author?
- Is it someone who is really an expert on the subject?
- Do other sources refer to the author with respect?
- Is the biographical and contact information about the author supplied and does it check out when you cross reference it
- Is the writer sincere and genuine or are they trying to scam you into buying something or into giving them something?

When you use information from the internet and supply information about yourself on the Internet, beware of plagiarism (to take information from the internet and present it as your own work without acknowledging your sources) and piracy (illegal copies of e.g. music).

Do not give out your personal details like your age, address and telephone number that reveal where you live, go to school or your schedule. Be careful about who you accept as 'friends' online and understand and use the security and privacy settings on social networking sites this will limit which parts of your profile are visible to others. Do not be rude, use swear words or watch pornographic material on the web

Data mining is the process of analysing data and looking for patterns to establish relationships. The information on the internet can be used to use for marketing or make predictions about the future. For example someone who often buys airline tickets would most likely also need a hire car.

The existence of modern ICTs makes doing business with international customers easier and cheaper. Advertising pays for search engines and other services on the web. People can now trade and share ideas without having to leave their own offices.

The use of ICTs as a tool in the political arena has in recent years become quite popular. For example President Jacob Zuma launched an online discussion of the ANC manifesto, on the internet and on cell phones. Another example would be US President Barack Obama's use of social media to get his message to more people during his presidency campaign.
This gap between those with regular, effective access to digital and information technology, and those without this access is called the digital divide. Africans experience several obstacles to ICT access e.g. poor infrastructure like telephone lines, unreliable power supply, poverty and illiteracy, especially since most content in the Internet is in English

One of the cultural benefits of computers and ICT is that they make it possible for us to accurately record and describe cultures so that we can learn about other cultures and help people to better understand each other. On the other hand, the spread of the predominant Western capitalist culture throughout the world, could lead to the loss of diversity.

Computers can cause many physical problems if you constantly and incorrectly use them, for example: 'Sore Thumb Syndrome' from typing on a cell phone. Typing on a cell phone involves making hundreds of tiny repeated movements that can lead to pain in the thumb.

The Social and Ethical Impact of ICTs

Repetitive strain is caused by repeated physical movements that can result in damage to tendons, nerves, muscles, and other soft body tissues. Other physical side effects are carpal tunnel syndrome and eye strain.

Psychological problems caused by ICTs include:

- People feel invaded and overwhelmed by technology. The feel like it has taken over their lives, intruding into their privacy, interrupting their conversations and concentration on other tasks. 'Multitasking Madness' or “Techno Stress” or “Information Fatigue Syndrome” are used when people are bombarded with more information than can be read and digested.
- People feel threatened and fear that they will lose their positions to younger more computer literate people. This makes them cynical and they resist change.
- Computer illiterate people tend to feel intimidated by technology and might feel their initiative, responsibility and decision making processes are being taken over by computers.
- Many people also suffer from what is called ‘Information overload’. These people are so overwhelmed by the volume of information available that it causes them to make poor decisions.

Addiction to social networks, games or specific types of websites or cell phones is increasing. Some of these individuals can’t work or study properly because they have a constant compulsion to check their phones. There are strong links between gaming and child obesity. It often leads to superficial communication that leaves no room for deeper, more thought out and meaningful communication or emotion. Computer games often idealise violence, crime, sexuality and materialism, which goes against the morals of many homes and religions.

Advances in computer technology have resulted in many unskilled jobs being taken over by computers, but on the other side, even more jobs were created for skilled labour. For example, designing, programming, installing, administering, trouble-shooting, fixing computers and manufacturing and selling computers.

The Social and Ethical Impact of ICTs

There has also been in recent years a move towards bringing work to people instead of people to work. It saves the company money, because they need less office space, and allows people to work flexi-time and adapt their work schedule to suit their family life or social schedule.
Disadvantages include: employees miss the personal contact with workmates and it is not always easy to keep work and family life separate.

Many technological processes deplete natural resources and produce unwanted by-products that pollute the environment. E-waste poses a significant risk to our personal health and environment. It is important that we recycle e-waste and dispose of unusable correctly in order to prevent this. As users we need to be more aware of the power requirements of devices when we purchase them, and try wherever possible to use technology which is more energy efficient.

Task

1. Name 3 toxic substances in a computer.
2. List 3 things that you can do to reduce the impact of technology on the environment.
3. Describe how can telecommuting help to reduce the impact on the environment

The Social and Ethical Impact of ICTs

Lesson 3 Cyber Threats and Safeguards to Protect you against Attacks

This lesson discusses various forms of malicious software, or "malware" like Trojans, Spyware and Key loggers as well as cyber-crimes like bandwidth theft, information theft, Hacking, Social engineering, Identity theft, Phishing, hoaxes and DOS attacks. The video even included an interview with a professional hacker! The last part of the lesson explains how you can protect your data by using up to date anti-virus and anti-spyware software, not to allow people you do not trust to use your computer, not to use public computers for your Internet banking or online shopping, making Backups, using firewalls and strong passwords.. You can also make use of digital signatures and digital certificates as well as data encryption to protect confidential information.

Lesson Outcomes

By the end of this lesson, learners should be able troubleshoot:
Distinguish between cyber threats like Spyware, key loggers, phishing, identity theft, social engineering, hacking, Internet-related fraud scams, hoaxes and DoS attacks.
Describe the purpose of safeguards like backup, Firewalls, Passwords, user access rights and privileges, digital signatures, digital certificates, encryption, SSL and Public Key encryption

Curriculum Links

Learning Outcome 2 and 3

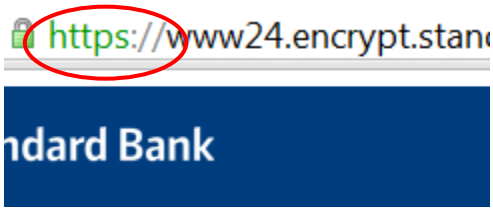
Assessment standards:

12.2.06 Critically assesses Internet security and its implications.
12.3.02 Analyses the effects of the use of computers across a range of application areas.

Lesson notes

You have to be able to describe the malicious software, "malware" or Cyber threats, and distinguish between them.

- Trojan – software that appears to be installing something that might for instance make your system better or faster, it is actually installing a completely different program onto your computer without your permission.
- Spyware - software that enters your computer as a Trojan and records or 'spies' on things ranging from the specifications of your computer to recording the websites you visit and sends that data back to its creator.
- Key loggers are a more sophisticated kind of software that records the keys you press on your keyboard or mouse clicks you make and even may capture screenshots of your computer. You can also get hardware key loggers. These types of devices work the same way as the software versions, but require that the person using the device have physical access to your computer to plug it in and retrieve it.



- Bandwidth theft and information theft is also fairly easy for people who have the know-how.
- Hacking is the process of accessing other computer systems from your account or computer without authorisation to send out spam or to gain access, steal or tamper with data and information on systems.
- Social engineering is a term used to describe activities that attempt to con and deceive people into giving out confidential information that will compromise the security of data or give access to a network.
- Identity theft is the crime of obtaining the personal or financial information of another person for the deliberate assumption of another person's identity, usually to gain access to that person's finances or to frame a person for a crime.
- Phishing is a form of Internet fraud where fake e-mail is used to try to get you to click onto a link for a website that looks official, like your bank's website for instance, but isn't the real thing. If you look closely at the fake email you will usually find a generic greeting and a request to click on a link.
- A hoax is a deliberate attempt to dupe, deceive or trick an audience into believing, or accepting, that something is real, when in fact it is not. Often the sender simply wants to generate a reaction or waste your time for their own amusement.
- A flooding denial of service or DOS attack, is a type of cyber threat that is generally designed to deny access to a specific site or network, either temporarily or permanently. A DoS attack sends real-looking but unnecessary messages to a victim computer. As a result, network bandwidth is wasted, disk space is filled with unnecessary data, or processing power is spent for useless purposes to prevent legitimate users access to system resources by shutting down or seriously slowing down a service provided by a computer system.
- A botnet, also known as a zombie army is a number of Internet computers that have been set up to forward transmissions like spam or viruses to other computers on the Internet according to the wishes of some master spam or virus originator. The owners are usually unaware of it. They are targeted if they fail to provide effective firewalls and other safeguards.

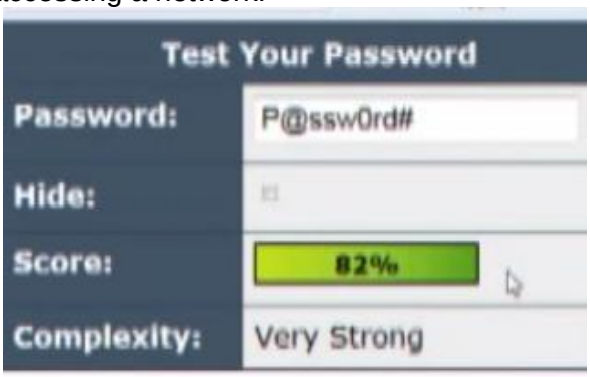
You need to protect yourself from these types of attacks. There are several ways that you can do this:

Make sure that you have up to date anti-virus and anti-spyware software installed on your PC. An anti-virus is protective software designed to defend your computer against malicious software. To be effective, your antivirus software needs to run in the background at all times. Because new malware crops up all the time, It is important to update your antivirus regularly so it recognizes new versions of malicious software. Do not allow people you do not trust to use your computer or install software on it! Ask technicians you don't know for identification.

You should also be aware of so-called shoulder surfers who will watch you when you type in a password or username and so-called dumpster divers who will go through rubbish bins to search for confidential information. Do not use public computers for your Internet banking or online shopping since you never know whether key loggers have been installed there.

The Social and Ethical Impact of ICTs

Back up your computer files regularly. To Backup means to copy files to another computer, tape, drive, or to server for safe storage. It protects you in case your computer is stolen or destroyed or the hard drive crashes. You can schedule Backups so that they are made daily. Using a firewall is also a good way to protect your computer from threats. A firewall is either hardware or software, or in some cases both. It is designed to, prevent unauthorized users or software from accessing a network.



Always use strong passwords, because they are harder to crack. A strong password is at least seven characters long. It doesn't contain a complete dictionary word, your user name, real name, or company name. It should also be significantly different from previous passwords and contain characters from uppercase letters, lowercase letters, numerals and symbols found on the keyboard. Never share passwords with anyone and change passwords immediately if they may have been compromised.

If you are sending data to someone else, you can protect the integrity of your data by using a digital signature. A digital signature is digital code that can be attached to an electronically transmitted message that uniquely identifies the sender to guarantee that the sender is who he claims to be. This signature is electronic and cannot be forged. When you are surfing the net you can check the integrity of the data on a website by looking for a digital certificate. Digital certificates are issued by a trusted third party called a Certificate Authority like VeriSign or Thawte. If you want to ensure your privacy or protect important data from unauthorised access you could make use of data encryption. Encryption is simply a way of scrambling text or data into a new format using a specified set of rules, making it impossible for unauthorized persons to understand. You can check if a web page is secure, by looking for a lock symbol somewhere on the browser and checking if the address starts with https. With Public Key encryption the message has two keys, or passwords. One key is used to encrypt the message - and another completely different key is used to decrypt it. Networks are also protected by network administrators who create user accounts and assign rights and permissions. On NTFS file systems on Windows computers, you can set security permissions on files and folders. These permissions grant or deny access to the files and folders.

Task

1. How do you back up the files on your computer?
2. Why is it important to update the antivirus daily?
3. Give an example of a strong password
4. Name an example where SSL with public key encryption is used.

The Social and Ethical Impact of ICTs

Lesson 1: The Impact of information and Communication Technologies on our Lives

	Advantages	Disadvantages
a. Virtual school	• can work at your own pace • save time and costs since you do not have to travel to school every day • Expert teachers can help students in rural areas where good schools are not available	• Lack of social contact with friends • Need self-discipline to work when a teacher does not check your homework regularly • Cannot ask questions as easily as in class
b. E- banking	• Available 24/7 • save time and costs since you do not have to travel to the bank	• People without internet access are excluded • Phishing attacks are used to steal your money online
c. Shopping	• Can compare prices • Can find goods that are not available near your home • Convenience if goods are delivered at your door	• Goods can look better in a catalogue that what it actually is • Added cost for delivery and packaging • Scams where you pay and never get what you ordered

Lesson 2: How Technology can Benefit or Harm the User, the Environment and Society

1. Lead, cadmium, mercury
2. Possible answers:
    a. Recycle batteries, printer cartridges
    b. dispose of unusable e-waste in special bins provided
    c. try wherever possible to use technology which is more energy efficient.
    d. use of power management systems which automatically turn off components such as monitors and hard drives after set periods of inactivity.
3. Tele-commuting and tele-presence technologies can help us to reduce greenhouse gas emissions related to travel since we do not have to go to work every day

Lesson 3: Cyber Threats and Safeguards to Protect you against Attacks

1. Click control panel, System and security, Backup and restore and follow the wizard
2. The antidote for new viruses must be downloaded to be protected
3. *P@ssw0rd#
4. Internet banking and online shopping

Troubleshooting

Series overview

There are 5 lessons in this series. They are structured and presented to give learners an understanding of the kinds of problems that can occur with computer hardware and software. The series takes a troubleshooting approach, attempting to uncover the cause of the problem through step by step questioning. Where possible, solutions to the problem are demonstrated practically.

Curriculum links

These video lessons achieve the outcomes and outcomes of the NCS, namely:

Learning Outcome 1: Hardware and System Software

The learner is able to demonstrate an understanding of and competently operate computer-based technologies.

Educational approach

The series shows learners how to work through systematically through a number of possible causes of hardware or system software problems, and how to select the best possible solution. The theory is supported by decision charts and discussions about what may or may not be wrong. Several solution options are presented and demonstrated, where possible,

Learners should be encouraged to try work out causes and solutions for themselves. They can also be guided to apply the learning demonstrated in this series on their own hardware or software versions. The division of lessons into distinct categories, such as hardware, peripherals, software and so on, makes it possible for learners to dip into these lessons when and where they need them, to learn more about problems they may be experiencing, or to deal with discrete unit of learning – depending on how the teacher would like to use these lessons. The lessons can be used in groups or as self-study units.

The lessons in the Troubleshooting Hardware and System Software Problems series follow the same approach, referring to a basic ‘model’: to diagnose what is wrong, follow a step-by-step process of elimination; collect information by asking questions; try to identify the possible problems and suggest solutions. It is important that learners apply what they learn to new scenarios and different data sets, and educators are encouraged to create examples of other problems or abnormal situations for learners to troubleshoot.

Troubleshooting

Free your brain

Series at a glance

Lesson title	Lesson Outcomes By the end of this lesson the learner should be able to:
1. Troubleshooting inside your computer	• Troubleshoot the power supply • Troubleshoot motherboard faults • Troubleshoot hard disk drive faults

2. Troubleshooting computer components and devices	• Identify and propose solutions for common problems with:
	  ○ The speakers, keyboard and mouse
	  ○ The monitor
	  ○ The printer

3. Troubleshooting software	• Hardware and system software faults like
	  ○ Common Operating System errors
	  ○ Problems with Utility software and
	  ○ Problems with Application software.

4. Troubleshooting networks	• Network Hardware Related Issues such as:
	  ○ Network device failures or malfunctions like networked printers, cabling, routers and modems
	• And Network Software-Related Issues, like:
	  ○ Using the Operating System
	  ○ Command-line troubleshooting
	  ○ Using Control Panel applets

5. Database Queries	• Identify and solve:
	  ○ Problems that occur with web browsing
	  ○ With e-mail software

Troubleshooting

Teaching Guidelines

Lesson 1: Troubleshooting inside your computer

In this lesson, learners will see various things to do to work out if the power supply is giving problems, and if so, learn some simple things to do to fix the problems themselves. The lesson also covers ways to check various aspects concerning the motherboard and hard disk drives. Learners are shown how to change a power supply and CMOS/RTC battery. If replacing the battery, remember to reset the CMOS values to the default setting. Hard disk problems could be a hardware or software problem.

Task

Learners must offer suggestions about the possible cause and solutions for the error message; “Invalid Media Type reading drive C”.

Lesson 2: Troubleshooting computer components and devices

The lesson is about checking and fixing common problems with computer components and peripherals. Many of the things that go wrong with speakers, the keyboard or the printer for example, are quite harmless and are usually easy to fix. This lesson guides learners to find the problems through a process of elimination, and discussed ways to fix many simple faults with devices.

Task

Learners must offer suggestions about the possible cause and solutions for the error message; “Invalid Media Type reading drive C”.

Lesson 3: Troubleshooting software

This lesson looks at how to identify and solve common problems that occur with the software installed on a computer. The lesson includes software issues like operating system, utility software and application software problems.

Task

Learners have to explain virtual memory and some problems associated with it, in the light of the message: ***Warning, the system is running low on virtual memory.***

They are also asked to provide solutions for these problems and to explain why these steps will sort out the virtual memory problem.

Lesson 4 Troubleshooting Networks

This lesson covers ways to identify and solve common problems that occur when connecting computers in a network, including software-related network errors and hardware-related network problems like failures in networked printers and issues with cabling, routers and modems.

Task

Once again, the task ask learners to provide reasons for and solutions to a practical, every-day problem.

Error 741: The local computer does not support the required encryption type.

Troubleshooting

Lesson 5 Troubleshooting the Internet

The lesson deals with troubleshooting the internet, including problems that occur with web browsing and with e-mail software.

Task

The task allows learners to put learning into practice for specific internet-related problems, namely that the internet page does not behave correctly and the formatting is messed up or just doesn't work properly. Learners must provide causes and viable fixes.

Troubleshooting

Lesson 1 Troubleshooting inside your computer

This lesson is about identifying and solving common problems that occur inside a person's computer. The lesson looks at things like desktop computer motherboard problems, hard drive faults and desktop computer power supply issues.

Lesson Outcomes

By the end of this lesson, learners should be able troubleshoot:

- The power supply
- Motherboard faults
- Hard disk drive faults

Curriculum Links

LO 1: Hardware and System Software

Lesson notes

Computer users experience common and relatively minor problems with their system software or hardware. It's not possible to solve all computer problems, but by practising some of the easier troubleshooting challenges, you will gain experience and often be able to solve bigger problems later.

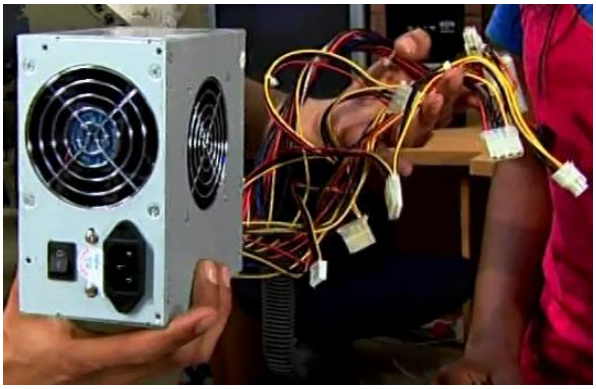
It's best to use a step-by-step process of elimination, to diagnose or work out what is wrong with a computer. Collect information by asking yourself questions. Then try to identify the possible problems and possible solutions.

Power Supply

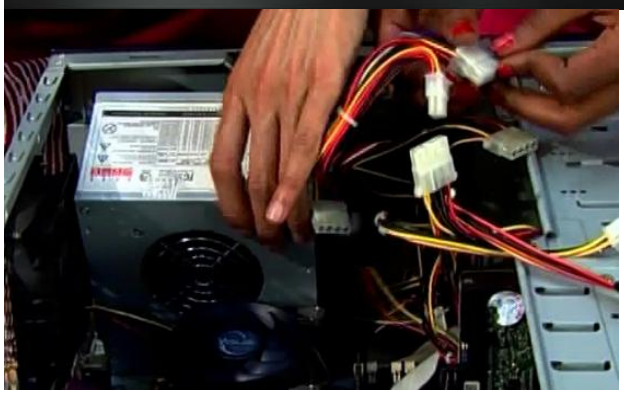
The first problem that we present in this lesson is a computer that is 'dead'. It doesn't seem to turn on, show any lights or open the screen. To eliminate things that are not wrong, we check that:

- the monitor is on
- the main switch is on
- the wall socket is working properly
- there is not a general power failure (plug in a lamp to check)
- the surge protector is turned on
- the connectors are connected properly

All these things are in order, but when the computer is turned on the fan doesn't work. This, and the fact that all the above items have been checked, tells us that the problem may be caused by a faulty power supply. You can buy and install a new power supply or take the computer in for repairs.

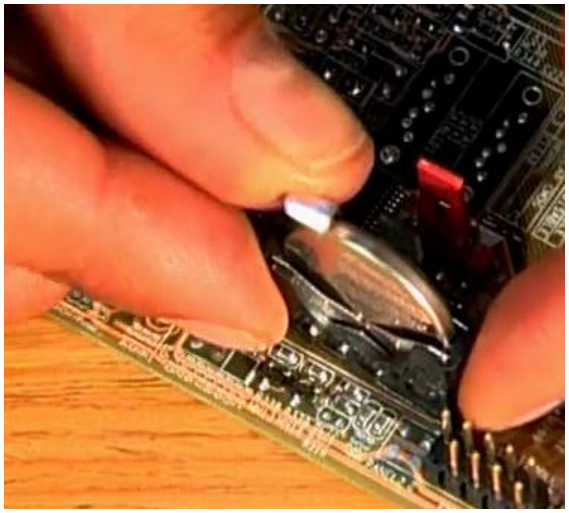


Troubleshooting



Motherboard

When checking for problems in the motherboard, always inspect the chipsets (Northbridge chip and Southbridge chip), all the sockets and channels, and the CMOS/RTC battery. If you have to replace the battery, remember to reset the CMOS values to the default setting when you turn on the computer.



Hard Drive

A hard disk drive error is often something like not being able to see a large portion of the hard drive. This could be a hardware or software problem, such as:

- The hard drive's limitations
- CMOS not setup properly
- Software / setup issue
- BIOS issue / limitation - hardware issue

Task

Your computer at home is giving the following error message "Invalid Media Type reading drive C".

1. What could be the cause?
2. How would you solve this problem?

Troubleshooting

Lesson 2 Troubleshooting computer components and devices

In this lesson, we learn how to check and fix common problems with computer components and peripherals. Many of the things that go wrong with speakers, the keyboard or the printer for example, are quite harmless and are usually easy to fix. This lesson will help you to find the problems through a process of elimination, and will help you to fix many simple faults with devices.

Lesson Outcomes

By the end of this lesson, learners should be able to:

- Identify and propose solutions for common problems with:
  - The speakers, keyboard and mouse
  - The monitor
  - The printer

Curriculum Links

LO 1: Hardware and System Software

Lesson notes

Speakers

When you're not getting any sound from your computer, like a Skype call or a video, check the external speakers or the volume control for built in speakers. When checking external speakers, make sure that they're plugged in properly, that they're in the right port (audio port is usually green) and that the electrical cable is plugged into the socket. When checking the volume control, slide the volume indicated to the level required and make sure that the mute option is unchecked.

Keyboard

Sometimes the keyboard gets stuck or 'frozen'. If it's a wireless keyboard, it could be the batteries. Otherwise, follow a list of things to check, like this:

- Check the cable connecting the keyboard to the computer
- Go to Start on the task bar and Shut Down the computer from the pop-up menu. Then I unplug the keyboard connection from the port, make sure it is clean and dust free, reinsert it tightly, wait 1 minute and reboot.
- Press the Caps Lock key. If the keyboard is working the Caps Lock light will blink on and off as you press it.
- If the Caps Lock light doesn't blink on and off, press each of the Shift keys 3 or 4 times and try the above step again.
- It might be your port. Try using any other USB port to connect your keyboard to your computer.
- Press Control Alt and Delete at the same time to bring up the Task List. Select any program that says 'Not responding' and click the End Task button. Keep doing this until all tasks are ended. Then check the keyboard again.
- If none of these steps help you to find the problem, you probably need a new keyboard.

Mouse

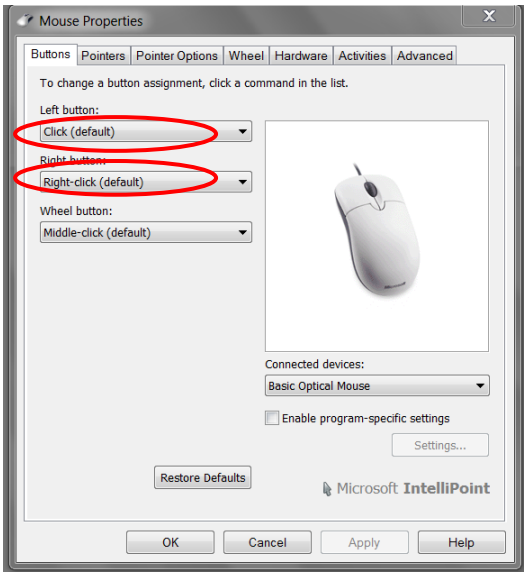
Perform various checks, such as:

• Is it a wireless mouse?	Check the battery
• Check connections	Disconnect and reconnect the mouse
• Are the connections firm?	Check that mouse is in correct port
• Is the program hanging?	Press Ctrl, Alt, Del once, choose end task, re-open program

Troubleshooting

- Also check the configurations of your mouse

:



Monitor

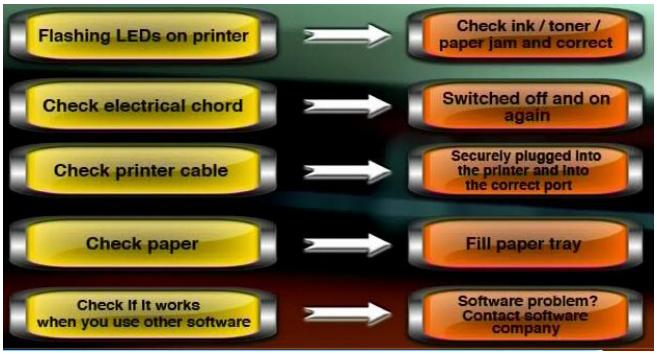
When the monitor is blank or malfunctioning in any other way, check that:

- The computer is plugged in and switched on
- The monitor switch is also on (light comes on)
- The computer is perhaps in sleep, suspended or hibernate mode by pressing Enter or Ctrl key
- The graphics card is faulty or needs to be reseated. Open the computer case and push it firmly back into its port.
- There isn't a problem with the screen cable, which you'd know if the monitor has a tint of one colour
- The monitor's brightness or contrast control hasn't been turned down

Once the monitor is working, it is also possible to adjust display settings, such as orientation or resolution, from the Control Panel:

Troubleshooting

Printer



It's also possible to correct problems like:

- The printer prints compressed or garbled print → This is probably a problem with Windows configuration; a solution can be found at http://support.attachmate.com/techdocs/1103.html
- The printing is smeared or has fine lines on the page → The fuser may be dirty or broken. Get this replaced by a professional.

Task

Your printer refuses to print due to a paper jam.

1. What steps could you take to fix a paper jam?

Troubleshooting

Lesson 3 Troubleshooting Software

This lesson looks at how to identify and solve common problems that occur with the software installed on a computer. The lesson includes software issues like operating system, utility software and application software problems.

Lesson Outcomes

By the end of this lesson, learners should be able to identify:

- Hardware and system software faults like
  - Common Operating System errors
  - Problems with Utility software and
  - Problems with Application software.

Curriculum Links

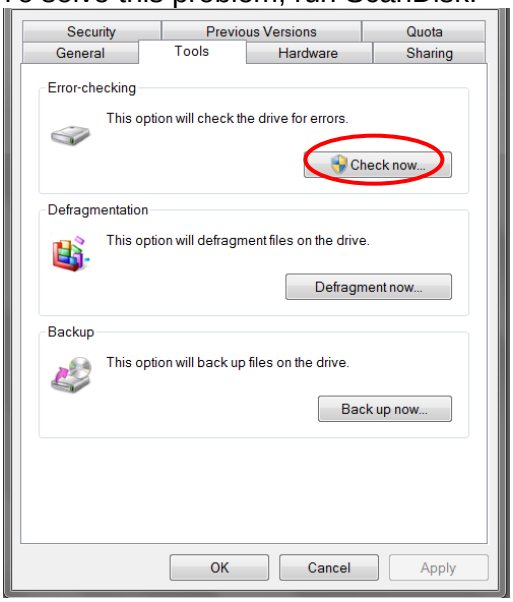
LO 1: Hardware and System Software

Lesson notes

Safe Mode

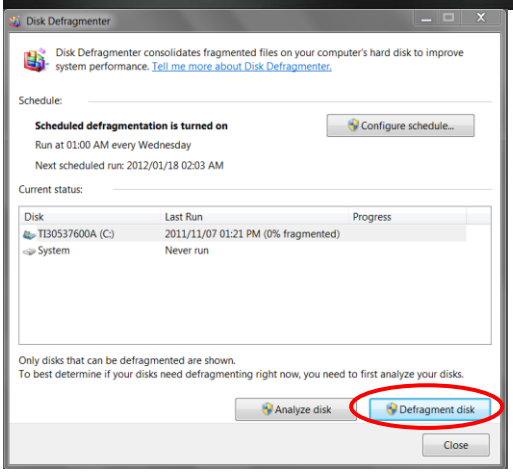
If the computer was not shut down correctly (i.e. turned off by pressing the power button), or if there was a power failure while using the computer, the computer will usually automatically open in safe mode. This is a limited version of Windows that loads when your PC is having problems.

To solve this problem, run ScanDisk.



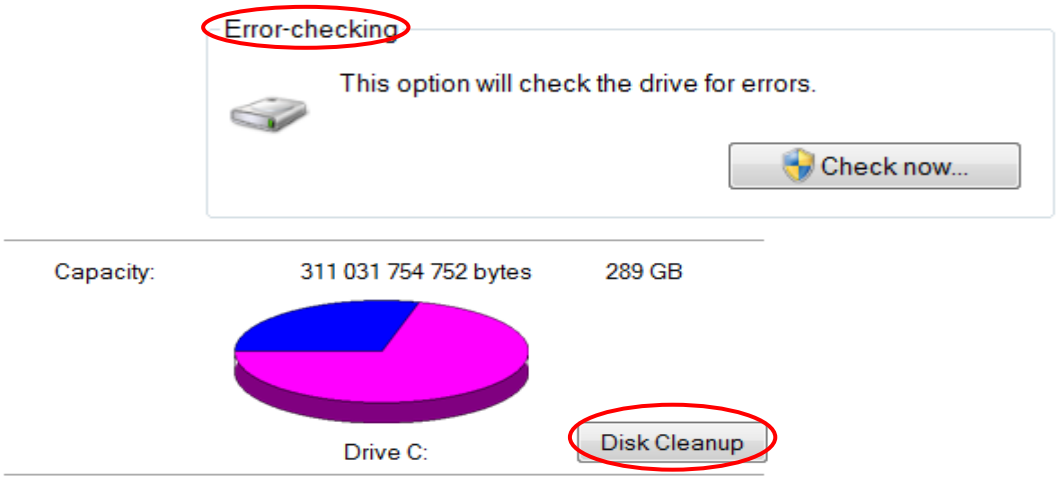
If the computer still opens in Safe Mode, run the Disk Defragmenter and reboot when complete.

Troubleshooting



Computer running slow

1. Many people find that their computer is slow; it sometimes shows that there's not enough hard disk space. There should be at least 200-500MB free hard drive space available. This allows the Swop file to increase in size and leaves room for temporary files.
2. Check that there's enough disk drive space; also run ScanDisk to make sure there's nothing physically wrong with the computer. And run Defrag to make sure the data is arranged in the best possible order.
3. Spyware and other malware cause many problems including a slow computer. Even if you have an anti-virus program installed run a malware scan.
4. Check for overheating. If your computer overheats, remove or disable any TSRs and start-up programs that automatically start when you boot. Also verify that the Device Manager has no conflicts.
5. If the problem still persists, check the utility software. Utility software helps manage your computer's
   hardware, operating system and application software. It includes disk defragmenters, system Profilers, virus scanners, application launchers, network managers and encryption utilities. The computer checks all this for you, and you can ask it to repair any bad sectors.



6. It's also a good idea to use the disk cleanup to free up space on your hard disk. It identifies files that you can safely delete and allows you to choose whether you want to delete it or not.

Installing Software

If you receive an error message when trying to install new software, these are some of the things to check:

- The software CD – it should be clean and scratch-free.
- The computer meets the minimum requirements for the software. Make sure the program is compatible with your operating system.
- If your computer runs out of disk space during the installation it would cause an installation error. Your best option then is to try and install it in safe mode.
- Verify the CDs are readable by reading the files on the drive. Use Windows Explorer, for example. If the CD attempts to AutoPlay, right-click the drive and click Explore to browse the drive.
- If there isn't enough disk drive space to install the software, run DiskCleanup and Defrag.
- If the CD-KEY or Serial Number gives you problems, verify you have the correct number. If you lost it or it does not work, contact the program developer.
- Sometimes a software program requires an update before it can be run on your computer. Check if there are patches or updates available from the developer.
- Next time you run it, make sure all the other programs are closed. If the program runs then, it's possible it has issues with other programs.
- Also verify your computer has the correct date. In some cases a program may rely on the date and if that date is incorrect it may cause issues.

Task

Every now and then the PC displays the following message**:**

Warning, the system is running low on virtual memory.

1. Explain what virtual memory is and how it functions.
2. Give TWO problems associated with virtual memory.
3. Provide solutions for the TWO problems indicated in 2 above.
4. Also explain why each of these proposed solutions will solve the

Troubleshooting

Lesson 4 Troubleshooting Networks

This lesson covers ways to identify and solve common problems that occur when connecting computers in a network, including software-related network errors and hardware-related network problems like failures in networked printers and issues with cabling, routers and modems.

Lesson Outcomes

By the end of this lesson, learners should be able to identify:

- Network Hardware Related Issues such as:
  - Network device failures or malfunctions like networked printers, cabling, routers and modems
- And Network Software-Related Issues, like:
  - Using the Operating System
  - Command-line troubleshooting
  - Using Control Panel applets

Curriculum Links

LO 1: Hardware and System Software

Lesson notes

A network is a group of interconnected computers and peripherals that can share software and hardware resources between many users, and the Internet is a global network of networks.

Connecting to the Internet

If you can't get into the Internet, check that you are using the right user name and password, and remember to check that caps lock isn't on. If using a router, make sure all the cables to the router and the computer are secure. Also check the port LED; if it is not lit, move the cable to a different port on the router. If all the cables are fine and all the LEDs are on, then power off and reset the router, wait 30 seconds, put the router on again and check the lights.

If the network activity LED lights are blinking, and you still can't log on, check that you're not logged on anywhere else - most users cannot have multiple logins. If this is the case, log off on one computer before you can log in with another.

Most small business, school and home computers use unmanaged hubs and switches. These are simple devices that act as a central connecting point where your Ethernet devices can talk to each other. Issues with unmanaged hubs and switches are usually caused by connectivity problems. If you can see some computers on the network, the hub or switch itself is probably operating properly.

Check connectivity: If other computers are visible on your network, troubleshoot the connectivity of the computers that are not visible:

- Check that the hub or switch is plugged in and the power light is lit.
- If a port light is blinking, change ports or reconnect the cable.
- Or change the speed of the network adapter card.
- If a link light is still blinking, reset the hub or switch. Replace cables with known good cables.
- Or remove the hub or switch by connecting two computers together with a crossover cable.
- Lastly check your model's Support product page. Maybe you need a software or firmware upgrade.

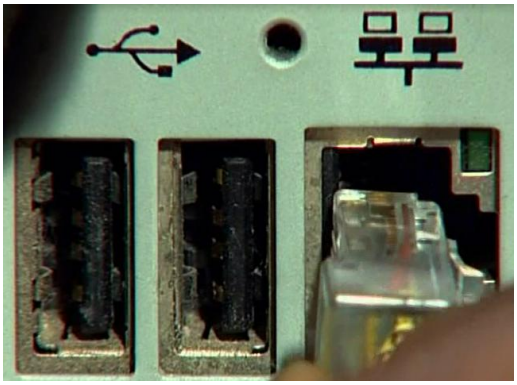
Troubleshooting

Remember, if you do not see any drives other than A, C and D drives, you are not logged into the network. Close all programs, log off, shut down and restart.

Connecting to Windows

Check the network cable, and then make sure the LED lights are lit up at the Ethernet port. This is where the Cat5 cable is connected to the computer. The lights should be blinking, showing that the network is active. If not working, try another network jack or another Cat5 cable.

There could also be a problem with your network card or that is not properly connected. A network card with a solid green LED or light usually indicates that the card is either connected or is receiving a signal. Flashing green light - It usually indicates that data is being sent or received. If the card does not have any lights or has orange or red lights, it is possible that either the card is bad, the card is not connected properly, or the card is not receiving a signal from the network.



Software Solutions

You can resolve many network and Internet connection problems by using software like the Operating System. Windows 7 has a Diagnose and Repair function. To find it go to Start and Control Panel. Through Network and Sharing, you should be able to find Diagnose and Repair. You will get a screen which will show that it is trying to identify the problem.
You will get a message if it cannot find a solution. If a problem is found, Windows Network Diagnostics will show the description and propose a solution, if possible.

You can also do command-line troubleshooting. The command line interface in Windows is accessed by typing CMD into the run line in XP or the search field in Vista. After typing in CMD click
OK or hit Enter. You can try pinging your network card to make sure it is functioning by pinging the local host. To ping the card or the local host type either ping 127.0.0.1 or ping local host when prompted. If you receive a reply 4 times, everything is in order. If not, investigate your NIC or Network Interface Card. Another easy way to find out if you are communicating to the Internet is to ping a known site such as Google or Yahoo.

Task

Your brother received the following error message:

Error 741: The local computer does not support the required encryption type.

Troubleshooting

Lesson 5 Troubleshooting the Internet

The lesson deals with troubleshooting the internet, including problems that occur with web browsing and with e-mail software.

Lesson Outcomes

By the end of this lesson, learners should be able to identify and solve:

- Problems that occur with web browsing
- With e-mail software

Curriculum Links

LO 1: Hardware and System Software

Lesson notes

Browser won't open

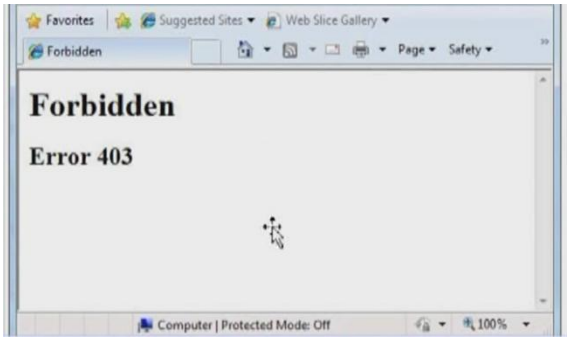
If your Browser doesn't open, reboot your computer. Many problems can be resolved that way. Then check for malware, viruses, and spyware.

Try a different home page: Select Internet Options . In the window that appears, under the General tab, change the Home Page, and try opening your browser again. If that doesn't work clear all cached data. Go to Internet Options as before, look under the General tab. Under Browsing History, click Delete. Click Delete Files and confirm by clicking Yes. Click Delete Cookies and confirm by clicking Yes. Click Delete History and confirm by clicking Yes
If this doesn't work, try returning your browser to its default settings in the same window.
If none of these solutions fix your problem you may need to uninstall and then reinstall the browser, or do patch updates.

Troubleshooting web browsing

The 2 biggest Internet browsers, Internet Explorer and Mozilla Firefox, will often prompt you when a plug-in is required. In most cases, it'll also give instructions on how to install them.

Sometimes, clicking on plug-in link on the site gives a 403 Forbidden Error Message. This error is an HTTP status code that means the page or resource you want to reach is absolutely forbidden for some reason. The lay-out of the error message differs, but the error code is usually clearly visible.



They might even provide a link to the site where you can download the plug-in.

Double-check that you typed the correct URL address. If that doesn't work you may need to contact the site owner for clarification.

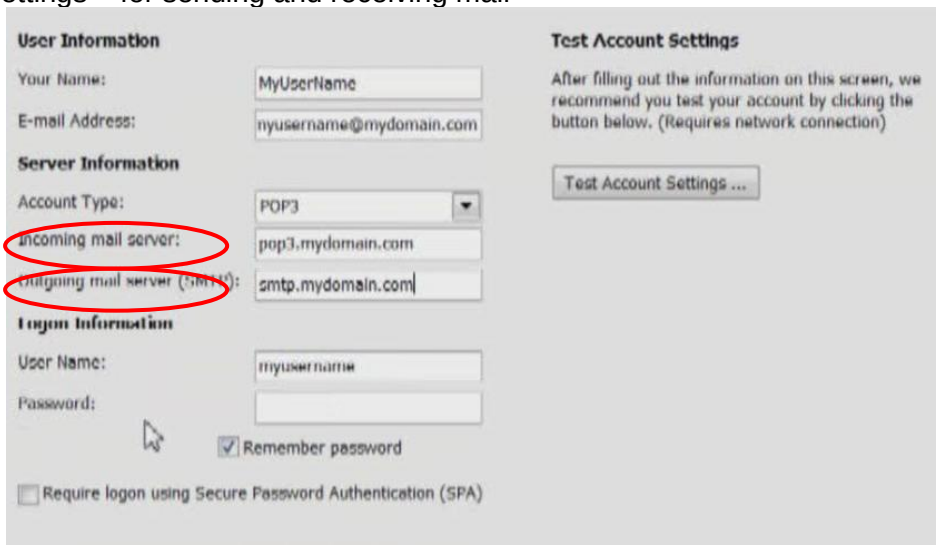
Troubleshooting

Another error message you could get when you use the Internet is the 408 Request Error Message. If you get this message, click the refresh / reload button in your browser.
If you still get the 408 message, try to access a common website like Google or Yahoo.
If those pages load normally, it means the 408 Request Timeout error only happens on the website you tried to access. You may need to visit the website at a later time, since it may be a web traffic problem. If Google or Yahoo doesn't load either you may have reached your ASDL cap, the total amount that you may download.

A common error is the 401 Unauthorized Error Message. When this happens, check the URL address, which must be correct. If that doesn't help visit the website's main page and look for a link that says Login or Secure Access. Register if needed, then login. If the page does not require authorization, the 401 Unauthorized error message may be a mistake. Email the owner of the website.

Problems with e-mail

If you have problems with e-mail, first do standard things like rebooting, check that you are using the correct username and password, and that connections are all in place. Now check if you can browse the internet. If none of these options work, check your Outlook or Outlook Express Settings – for sending and receiving mail



Troubleshooting

If the problem continues even after checking mail settings, login to your Web Mail and Delete any suspicious email that can block your Mailbox. Corrupted spam emails can cause Outlook or Outlook Express to hang, as well as emails that have no “From” address.

If you get an error message for e-mail, write down the message and number. Example: **error number 0x800ccc15… Copy and paste it to** Google or Yahoo or any other search engine. You will find hundreds of results and solutions to your specific issue.

Task

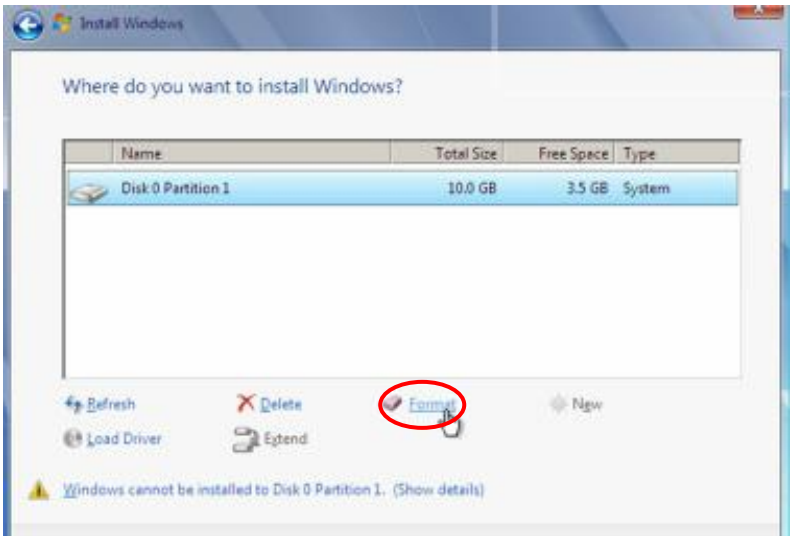
An internet page you’re trying to view does not behave correctly. The formatting is messed up or it just doesn't work properly.

1. What could cause this?
2. What could you do to fix the problem?

Troubleshooting

Lesson 1: Troubleshooting inside your computer

1. The hard disk drive does not have a valid format, or could be faulty.
2. You will need to reformat the computer. After backing up all data, restart the computer and insert the Windows CD into the CD drive while the computer is restarting. Press enter when asked if you want to reboot from the CD, and wait for all the Windows files to load. To start the wizard, enter language options and a few details, select the drive to format, and go to Options; select Format and continue with the wizard. You can also choose to Repair your Computer to fix system errors or startup problems.



Lesson 2: Troubleshooting computer components and devices

- Check the LED lights on the top of the printer and see if an error light is lit.
- Some printers may point you to where the printer jam exists. If not, you may need to investigate yourself.
- Remove the toner cartridge, so you will have easier access to the feeder and where the paper has probably jammed.
- Gently remove the jammed paper, try to avoid ripping the paper, since this will make unjamming the printer more difficult.
- Once the jam has been removed, the printer should be able to try to print a job again.
- If not, try turning the printer off and on again.

Lesson 3: Troubleshooting computer components and devices

1. Virtual memory is the hard drive space used when there is not enough RAM for the programs you want to use. It's like extended memory. Virtual memory moves inactive items, applications and data, to a location on the hard drive for a while; if you need to use this application and data later, the computer moves it into RAM and moves something else to virtual memory.
2. Two problems with virtual memory are that your computer may not have enough virtual memory (giving the message "the virtual memory is too low") or it may need more RAM.
3. Increase virtual memory size on your computer (changing the settings under Advanced Performance Settings) or purchase additional RAM if increasing virtual memory does not help. Remember, to work well, Windows 7 needs about 1 GB of RAM to run, Windows Vista needs 512MB RAM and Windows XP needs 256MB; when certain applications (like gaming, for example), more RAM will be needed for all these operating systems.

4. Increasing the virtual memory settings will give more hard drive space to which the computer can move applications and data. Increasing RAM will give more operating memory, so that virtual memory may not need to be used.

Lesson 4: Troubleshooting Networks

1. The fix the 741 error, download and install these software tools (Windows Software Update and Error Code 0x2E5 Fix. In some cases, error 741 is caused by spyware or viruses, so run virus checks and spyware removers.

Lesson 5: Troubleshooting the Internet

1. Each browser version has changes to the way HTML and other code is interpreted. This is usually done to become more standards compliant or because some things were removed. Microsoft Internet Explorer 8 compensates for this with the Internet Explorer 7 compatibility view feature. You may have compatibility view turned on, which tries to render the site in IE7.
2. This can usually be corrected by selecting Tools and Compatibility View within Internet Explorer 8. Or go to internet Options, select Advanced and then Browsing. Uncheck 'Recover from page layout errors with Compatibility View'.

END