

NATIONAL CERTIFICATES (VOCATIONAL)

GUIDELINES FOR THE INTERPRETATION AND ASSESSMENT OF

COMPUTER PROGRAMMING

NQF Level 4

June 2009

ASSESSMENT OF SUBJECT OUTCOMES IN COMPUTER PROGRAMMING - LEVEL 4

Topic 1: Object Oriented Programming

SUBJECT OUTCOME	
1.1 Describe basic object oriented terminology	
ASSESSMENT STANDARD	
• The description explains the basic principles of a class.	• Explain the basic principles of a class. • Theoretical concepts applied into real world problem solving and program development. • Concepts explained via concrete practical programming and or UML examples o ADT o Abstraction o Class o Object o Inheritance o Polymorphism o Properties o Attributes o Methods Encapsulation
• The description explains the basic principles of an object. Range: Definition and implementation of objects	
• The description explains the basic principles of information hiding and encapsulation.	
• The description explains the basic principles of inheritance	
• The description explains the basic principles of polymorphism	
ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/practical work • Assignments • Group work • External Exam - Theory questions	

SUBJECT OUTCOME	
1.2 Describe the fundamental differences between object oriented and procedural programming	
ASSESSMENT STANDARD	
• The description explains the implementation of classes in object orientated programming, using examples	• Explain the implementation of classes in object oriented programming, using examples Contrast: Functions Procedures and methods Abstraction
• The description explains encapsulation of data and functions/methods in OO versus procedural programming	• Procedural and OO programming are contrasted, strengths and weaknesses of each
• Global data sharing is minimised to enable weak coupling, and modules exhibit functional cohesion.	• Explain encapsulation of data and functions/methods (in classes)
• The description identifies possible classes for simple examples and problems *Range: similar complexity, to application mastered in Basic programming principles and standards*	• Describe how global data sharing is minimised to enable weak coupling. • Describe how modules exhibit functional cohesion. • Explain and identify possible classes for simple examples and problems

	KEY WORDS, CONCEPTS and APPLICATION • Design a class using UML from a given scenario • UML Class diagrams interpreted into program code. • Methods – Accessors and Mutators • Properties • Attributes • Constructors • UML Class diagrams are used to illustrate relationships and inheritance. E.g. Composition vs Inheritance • Methods and attributes are identified • Virtual and Abstract Methods • Overridden Methods • Polymorphic Methods • Program code is written from a designed class. • Difference between by Value and By Reference Parameters •

ASSESSMENT TASKS OR ACTIVITIES

- Class tests
- Projects/practical work
- Assignments
- Group work
- External Exam - Theory questions

SUBJECT OUTCOME	
1.3 Implement object oriented techniques in development of a solution	
ASSESSMENT STANDARD	
• Objects and classes are identified and classes are defined to provide a solution to a given problem/scenario • *Range: object identification, class design, objects instantiation, object array/s, methods, encapsulation, inheritance, polymorphism*	• Identify objects and classes • Implement a class/classes in a solution to a given practical problem/scenario **KEY WORDS, CONCEPTS and APPLICATION** • VB Classes and Classes and objects (User created and. NET Library) • Develop a class with multi-datatype attributes e.g. strings, dates, etc. • VB.NET Solutions are developed (coded / programmed) implementing programmer defined classes with a strong focus on problem solving and logic. • VB.NET GUI Solutions are designed and developed and coded which include: - Class declaration – attributes and methods - Create object - Instantiation - Encapsulation - Contructor methods - Accessor and Mutator methods

	- Single Inheritance - Composition - Polymorphism • Array of class objects are used within the VB.NET Solution. • Array objects are instantiated with data from a the user/ text file / database • Input into the Object • Array of objects are manipulated • Operations on and manipulation of array of objects. - Tracking the number of elements in an array - Inserting elements - Deleting elements - Sorting - Processing: sum, average • Data from an object is saved: Textfile, DB • Text files: (Text File Objects) - Linking to a text file - Open, Close - Loop through text file - Import data from text file

	- Export data to text file • VB.NET Solution may span over several forms • Reports are created: Text based • Implementation of exception classes and error handling are implemented in the se designed solution: • Exceptions / Try / Catch • Defensive Programming

ASSESSMENT TASKS OR ACTIVITIES

- Class tests
- Projects/Practical work
- Assignments
- Group work

External Exam Theory	External Exam Practical
Student should be able to create a class, attributes and identity, methods from a given scenario in a diagram format. Inheritance and associate in a diagram format Difference between declare by VALUE and by REFERENCE	From a given class diagrams .design a program using VB.Net **Include:** • Objects • Methods • Functions

•

Topic 2: Database application design

SUBJECT OUTCOMES	
2.1 Review the requirements for database access for a computer programming solution.	
ASSESSMENT STANDARD	
• The review identifies and explains the feasibility of the requirements. • Client needs ability to provide for needs.	• Identify and explain the feasibility of the requirements. • Technical feasibility, input, output and processing requirements (6) **KEY WORDS, CONCEPTS and APPLICATION** • DB Design • Normalised tables • COST / COCOMO o effort =(system size)*(productivity rate) o other o algorithmic model o expert judgement o analogy o Parkinson o Price to win o Top-down o Bottom Up

• The review identifies the database access objectives and critical performance factors. • Motivate o why a database o what is needed o how is it designed o how do deliverables influence the requirements of the design	• Explain and identify database access objectives and critical performance factors 12. User rights 13. Security 14. Large database vs small database.(accessibility) (4)
• The review explains how to estimate costs for the development effort required. (cocomo)	• Explain how to estimate costs for the development effort required. • parametric models • effort =(system size)*(productivity rate) • other • algorithmic model • expert judgement • analogy • Parkinson • Price to win • Top-down • Bottom-up • Effort is a function of the system size combined with production rates (how much work someone can complete in a

	given time). • Cocomo model [4] designed by Barry W. Boehm. • Effort=1.4*thousands of lines of code
• The review explains the need for adopting a review procedure to ensure that the outcomes meet the database access requirements.	• Explain the need for adopting a review procedure to ensure that the outcomes meet the database access requirements (2) • Required: o Local database o Produced on paper/screen o What o Why o how
ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/Practical work • Assignments • Group work	

SUBJECT OUTCOMES	
2.2 Design database access for a computer application	
ASSESSMENT STANDARD	
• The design describes and demonstrates design implementation per user requirements (theory and application).	• Describe and demonstrate design implementation per user requirements (a scenario will be given and students are expected to identify simple dfd user requirements) (3) **KEY WORDS, CONCEPTS and APPLICATION** • Keys: Primary, Foreign, Combined Primary Keys, Indexes • DB Design • Tables • Relations: 1:1 1:M M:1 M:M are avoided and redesigned to 1:M – M:1 • Referential Integrity • Data Access SQL ADO Connections • Data Validation • Data Redundancy • Normalisation, 1st 2nd and 3rd • ERD are interpreted into the design of a DB

• The design of the database structure that resembles the output from the data analysis is described and demonstrated.	• Describe and demonstrate the design of the database structure that resembles the output from the data analysis. • Use diagrams to design a database structure and output requirements for a given scenario to conform to the input. • Describe concepts.
• The manner in which the design ensures that the structure of each table in the database adheres to the third normal form (normalised tables) is described and demonstrated.	• Describe and demonstrate how to ensure that the structure of each table in the database adheres to the third normal form and implement the steps required to normalize the database to the third normal form (6).
• The design identifies and demonstrates the methods of accessing data.	• Identify and demonstrate the methods of accessing data directly and indirectly.
• The key relationships between the tables within the database are identified. • Technologies involved. • Specifics as allowed by the specified language	• Identify the key relationships between the tables within the database 15. One to Many 16. Many to Many 17. One to One 18. The reasons for not using many to many (2)
• Strategies in the design to ensure that the data types for primary and foreign keys are consistent throughout the database are explained and demonstrated.	• Explain and demonstrate how to ensure that the data types for primary and foreign keys are consistent throughout the database.

ASSESSMENT TASKS OR ACTIVITIES
• Class tests • Projects/Practical work • Assignments • Group work

SUBJECT OUTCOMES	
2.3 Write program code for database access for an application implementing SQL	
ASSESSMENT STANDARD	
• A method for external data connection and access is identified, and implemented using program code • *Range: a connection string, data base object, data objects e.g. table objects, session objects*	• Identify and implement a method for external data connection and access using program code • Practical: designing a database and connect it to VB.NET using OLE DB • Use SQL statement to query the designed database **KEY WORDS, CONCEPTS and APPLICATION** • Access a database through a programming language • Setup an OLEDB connection or connect to database by providing path in VB.NET statements • MsAccess Database as designed -> OLEDBConnection [Connectionstring] • DBConnections, DataTables, DBCommands, DataAdapters, DataRows, SQL Objects

	• Create database solution using multi-tables in VB.NET • Data redundancy and Normalisation by inspection • Create a DB Application solution to a set of design specifications by inserting a programmer designed DB (MDF, data object, designed and created via the object inspector) • Use the DB Wizzards where applicable for the development of a DB Solution • Creating a Multiform DB Application • Develop a DB Solution with a MDI interface • Implement data aware components in the development of the solution e.g. DataGridView • SQL in a VB.NET Application - Display records from a table in a sorted order - Group related data together - Creating calculated fields - Formatting with round, int, etc. - Create a join query (single joins) using 'WHERE' - Mathematical operators - Special operators: BETWEEN, IN, LIKE - Aggregate function SUM, AVERAGE, MIN, MAX, COUNT

	- Date functions • Modify data in a Table using SQL – DML Statements - Delete records meeting specified criteria from a table using SQL Statements (DELETE) - Modify records and Update the content of certain fields in a table meeting specified criteria from a table using SQL Statements (UPDATE, SET) - Add records meeting specified criteria from a table using SQL Statements (INSERT) • Implement parameterised queries in the design of the solution where applicable • Add data to a database imported from a Sequential textile • Implement user defined classes as part of the DB solution • Use DB table objects to insert data: DataTable, Records. Etc • Validate data: Data validation, Exceptionj Handling, referential integrity • Develop the VB.NET DB Solution implementing code
• The manner in which the program code uses language constructs to facilitate the implementation of the solution is demonstrated.	• Demonstrate how the program code will use language constructs to facilitate the implementation of the solution. Connection strings required

• Refer to a program solution in VB.Net.	
• Tables are joined in a query to satisfy a requirement	• Explain and demonstrate how to join tables in a query to satisfy a requirement • Joining tables in a query and print them and the output (4)
• Program code such as SQL is constructed that preserves the integrity of the data being accessed by multiple users and processes. *Range: inserts, updates* • A description is provided on how	• Describe how to construct program code that preserves the integrity of data being accessed by multiple users and processes. Theoretical User right views
ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/Practical work • Assignments • Group work	

SUBJECT OUTCOMES	
2.4 Test programs for an application that accesses a database	
ASSESSMENT STANDARD	
• The manner in which testing checks all program logic paths is described and demonstrated.	• Describe and demonstrate how testing checks all program logic paths.

• Data captured in the correct form.	**KEY WORDS, CONCEPTS and APPLICATION** Data Validation Exception Handling • Operations of program conform to the design specifications. • Relative and Absolute paths Or • Javascripts, CCS, HTML (2)
• The manner in which testing corrects program code to eliminate errors identified is described and demonstrated.	• Describe and demonstrate how testing corrects program code to eliminate errors identified. • Comments in coding facilitate error corrects
• The manner in which testing verifies that the functions access the database in the required environment is described and demonstrated.	• Describe and demonstrate how testing verifies that the functions access the database in the required environment
ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/Practical work • Assignments • Group work	

SUBJECT OUTCOMES	
2.5 Document programs for a computer application that accesses a database	
ASSESSMENT STANDARD	
• The manner in which the documentation enhances the understanding of the program code is described.	• Describe how the documentation enhances the understanding of the program code. **KEY WORDS, CONCEPTS and APPLICATION** UML Classes ERD Comments User Manuals
• The manner in which the documentation complements self-documenting attributes of the program code is described.	• Describe how the documentation complements the self-documenting attributes of the program code.
• Documentation is developed to support the design, program code and solution	• Develop documentation to support the design, program code and solution (students are expected to develop a user my manual for a specific program designed, which will be applicable for ISAT)
ASSESSMENT TASKS OR ACTIVITIES	
• This Subject Outcome should be assessed only in the ISAT.	

Topic 3: Principles of developing software for the internet

SUBJECT OUTCOME	
3.1 Explain the network issues related to internet applications	
ASSESSMENT STANDARD	
• Session-less network protocol for the internet is explained	• Explain the internet in terms of a session-less network protocol. **KEY WORDS, CONCEPTS and APPLICATION** HTTP Bandwidth Session Less Web browser Server Response Time • Data Package Size
• The explanation lists the implications of session-less application development.	• List the implications of session-less application development.
• The impact of limited band-with on internet usage and data transfer is explained.	• Explain the impact of band-width on internet usage and data transfer • The lower the band-width, the slower the down load time.

	More users on the internet cause a delay in the data transfer • The more multi media intense the band width the higher the bandwidth requirement. • Multi media requirements include videos, graphics, animations and sound. • Bandwidth 'gulpers" • Explain how different browsers would render the same code.
• The implications of band-width for application design are listed.	• List the implications of band-width for application design. • Application that has more pictures will take time to load • The computer uses cache to store pictures that are down loaded. • Host company, service providers and end users • Hosting storage size and hosting band width
ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/Practical work • Assignments • Group work	

SUBJECT OUTCOME	
3.2 Explain the implications of copyright, ownership and royalties	
ASSESSMENT STANDARD	
• Copyright issues related to Internet development is explained	• Explain copyright issues related to internet development (2).
• Ownership issues related to Internet development is explained	• Explain ownership issues related to internet development (2).
• Royalty issues related to Internet development is explained	• Explain royalty issues related to internet development (2).
ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/Practical work • Assignments • Group work	

SUBJECT OUTCOME	
3.3 Explain version control and security issues related to internet applications	
ASSESSMENT STANDARD	
• Version control issues (compatibility) related to internet development is explained	• Explain version control issues related to internet development. • Internet Explorer versions • Versioning of browser • Versioning of browser type • Versioning of website • Versioning of language type

• Security issues related to internet development is explained • Hacking • Phishing • Cracking	• Explain security issues related to internet development. Passwords Fire walls Encryption
• Ways of managing security issues related to internet development is explained	• Explain ways of managing security issues related to internet development. Passwords Fire walls Encryption
ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/Practical work • Assignments • Group work	

SUBJECT OUTCOME	
3.4 Demonstrate the basic implementation of different user interface methods used for internet applications	
ASSESSMENT STANDARD	
• The demonstration identifies and explains different user interface methods used for Internet application development. *Range: web forms, server pages.*	• Identify and explain different user interface methods used primarily for internet application development. Different types of browsers Or Application like Dream weaver, front page. Homesite (3) **KEY WORDS, CONCEPTS and APPLICATION** Identify the different categories of scripting Explain the different user interface methods Demonstrate the basic implementation of the user interface methods Web Forms Server Side Scripting Applets
• The demonstration indicates the implications of each method.	• Indicate the implications of each method as listed above • Compare above listed tools and languages.

• The basic implementation of each of the user interface methods is demonstrated	• Demonstrate the basic implementation of a specific user interface method above. Different types of browsers Or Application like Dream weaver, front page. Homesite
ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/Practical work • Assignments • Group work	

Topic 4: Design and build a web-site using HTML

SUBJECT OUTCOME	
4.1 Explain basic guidelines for web-page design	
ASSESSMENT STANDARD	
• The nature and use of a web-site is explained.	• Explain the nature and use of a web-site. **KEY WORDS, CONCEPTS and APPLICATION** Identify the different types of websites and explain the different

	uses for each type Should be able to identify the different physical content on a web page. Should be able to explain the transactions that web pages can used for Website Static Website Dynamic Website Web Page Organized by function a website may be: - Personal web sites - Commercial website derives revenue by offering products or services. - Government websites - Websites for non-profit organizations. Post and Read • Web site elements
• The physical content of web-pages is identified and explained • Design for screen vs paper • Design for disabilities e.g. hearing & visual	• Identify and explain the physical content of web-pages
• Typical transactions which can be carried out via a web-page are explained.	• Explain the typical transactions which can be carried out via a web-page

ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/Practical work • Assignments • Group work	

SUBJECT OUTCOME	
4.2 Use core HTML to build a web-page	
ASSESSMENT STANDARD	
• The advantages and disadvantages of HTML editors and other web-site design tools are discussed.	• Discuss the advantages and disadvantages of HTML editors and other web-site design tools. • What are dream weaver, front page & home site? **KEY WORDS, CONCEPTS and APPLICATION** 1) Creating a HTML Page • What is XHTML • What is a Web Page • How does a web page work • Elements of a web page • Creating a simple web page - Create a webpage using Notepad - Tags that a web page must have

	- View webpage in a web browser • Web browsers - Internet Explorer - FireFox - Google Chrome - Viewing Web Page Source Code • W3C Validation 2) Elements of a web page • Text formatting and Font Control - Using Boldface and Italics - Font Size - Font Colour - Special Characters - Linking to other pages • Headings • Text Areas / Text Fields • Buttons • Radio buttons / groups • Checkboxes • Horizontal Rules • Line Breaks

	• Adding background colour
	• Lists
	  - Unordered Lists
	  - Ordered Lists
	  - Definition Lists
	  - Nesting Lists
	
	3) Web page design
	• Forms
	• Basic Frames
	• Basic Tables / Nested Tables
	
	4) Web page graphics
	• Images
	• Images Maps
	• Adding background images
	
	5) CSS

• Basic HTML functions are integrated in the design of a simple web page	• Integrate basic HTML functions in the design of a simple web page and elements/tags.

	• Identify and explain the physical content of web-pages <html> <head> <title></title> </head> </body> < a href> self referencing link # <Table</Table><tr></tr><td></td> h1,h2,h3 Spanning Columns, Spanning Rows <th></th> <form> Element <input> element Types(text box,Password,Check box,radio,submit,reset,hidden, image, button, file) <select> Element <option> <optgroup> <textarea> Element <fieldset> and <legend> CSS- Cascading style sheets

	Font Family, font size,Font Style,Font –Varient,Font Weight division (span of words) Class Attribute Selector Specifying Letter Spacing, Specifying word Spacing, Specifying Color values Style rule {margins, borders and padding}
• The HTML facilities that apply to typical web transactions are defined • Status line • Image maps • Mail forms	• Define the HTML facilities that apply to typical web transactions • Basic validation is required. • Fill in the form n Java Script. • Submit the form by mailing it.
ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/Practical work • Assignments • Group work	

Topic 5: Create multimedia, web-based applications with scripting

SUBJECT OUTCOME	
5.1 Plan the use of a multimedia, web-based authored application	
ASSESSMENT STANDARD	
• The user-specified topic, purpose, target audience and objectives of the application are identified according to agreed development plan	• Identify the user-specified topic, purpose, target audience and objectives of the application according to agreed development plan. A scenario will be given and students are expected to outline it according to the above criteria (4) **KEY WORDS, CONCEPTS and APPLICATION** Development plan is a document that consists of a: - Personnel schedule - Timetable for each phase of the software project - Cost schedule - Documentation schedule - Testing schedule. A target audience is a group of people for which an application is developed.

	Application objectives are the planned or intended outcome of an application. Acceptance criteria are the criteria for inspection specified by the client to be able to determine if the objectives of the development plan was met. Gantt chart is a type of bar chart that illustrates a project schedule.
• The way in which the tools selected to create multimedia, web-based applications with scripting are justified in relation to agreed development plan is explained.	• Explain how the tools selected to create multimedia, web-based applications with scripting are justified in relation to agreed development plan.
• The hardware and software required to create and run the application according to the agreed development plan are identified.	• Identify the hardware and software required to create and run the application according to the agreed development plan.
• The plan for the creation of a multimedia, web-based application is outlined and monitored according to project planning principles and financial requirements.	• Outline and monitor the plan for the creation of a multimedia, web-based application according to project planning principles and financial requirements.
• The configuration of the computer and associated systems necessary for the creation of the application according to the agreed development plan are described.	• Describe the configuration of the computer and associated systems necessary for the creation of the application according to the agreed development plan.

ASSESSMENT TASKS OR ACTIVITIES
• Class tests • Projects/Practical work • Assignments • Group work

SUBJECT OUTCOME	
5.2 Design a multimedia, web-based application	
ASSESSMENT STANDARD	
• Story-boards and flow-diagrams of the multimedia, web-based applications are designed to ensure effective communication between developer and user. • Simple use cases • Site map	• Design story-boards and flow-diagrams of the multimedia, web-based applications to ensure effective communication between the developer and user. Use Power point **KEY WORDS, CONCEPTS and APPLICATION** Storyboards Flow Diagrams UML Use Cases User specifications is a document that describes what the system should do, not (necessarily) how the system should do it.

	A UML Use-Case diagram is used to determine the functionality of a software application by highlighting the users and their intended actions to be taken. A Site map is used to show the overall structure and hierarchy of a Web site on which site navigation will be based A Wireframe is used to convey the general page structure and content requirements for individual pages. It does not specify any graphical detail like font colour and image sizes. Story-boards are screen shots or some type of graphical representation of the screens, combined with a narrative description. Storyboards help to document the functionality of the site and describe how users will use the interface.
• A multimedia, web-based application design is explained and motivated according to user specifications.	• Explain and motivate a design for a multimedia, web-based application according to user specifications. Draw a diagram to represent a story- board.

ASSESSMENT TASKS OR ACTIVITIES
• Class tests • Projects/Practical work • Assignments • Group work

SUBJECT OUTCOME	
5.3 Identify appropriate text, graphic elements and animation	
ASSESSMENT STANDARD	
• The manner in which multimedia, web-based application text, graphics elements and animation are aligned with agreed topic, purpose and target audience for the application and South African copyright and privacy laws, is explained and demonstrated.	• Explain and demonstrate how to align multimedia, web-based application text, graphic elements and animation with agreed topic, purpose and target audience for the application, considering South African copyright and privacy laws. **KEY WORDS, CONCEPTS and APPLICATION** Graphic elements, in terms of web development, can be defined as any static images visible on a web-page. Graphical elements, in terms of web development, can be defined as any visual component visible on a web-page.

	A Style guide is used to document baseline design requirements for a website. It usually defines font classes and a wide range of various design conventions to be followed. An Animation is a graphical element that changes its visual content over time. The File Format of a file refers to the way in which the binary data is organized in the file. Some file formats makes use of a technique called compression to reduce its size. Typical file formats on the world wide web are JPG, GIF, PNG MPG, MP3 DivX and FLV.
• The manner in which text, graphics elements and animation saved according to agreed design specification considering South African copyright and privacy laws is explained and demonstrated.	• Explain and demonstrate how text, graphic elements and animation are saved according to agreed design specification, considering South African copyright and privacy laws. (check the symbol)
• The manner in which text, graphic elements and animation are saved in a format that allows them to be integrated into the multimedia, web-based application is explained and demonstrated.	• Explain and demonstrate how text, graphic elements and animation are saved in a format that allows them to be integrated into the multimedia, web-based application. PNG,JPG or JPEG,GIF,SVG, FLW and Videos
ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/Practical work	

• Assignments • Group work

SUBJECT OUTCOME	
5.4 Create multimedia, web-based application scripts.	
ASSESSMENT STANDARD	
• The script is explained and demonstrated using a diagram	• Explain and demonstrate the script using a diagram • Explain the composition of a line of java script using a diagram **KEY WORDS, CONCEPTS and APPLICATION** A Multimedia website refers to a website that consists of text, images, animations and sound etc... A Scripting Language is used to reduce the size of files transferred over the internet. A script consists of a collection of scripting language instructions that requires additional software to be pre-installed on the client PC before it will execute properly. Java script is a scripting language that is based on the Java programming language. All browsers support java script. A Client side program is a program that is executed on the PC used to access the web site.

	Flow-diagrams are a graphic means of presenting an overview of how processes work, usually consisting of graphic boxes and other shapes, and lines and arrows. Error checking is a fault finding process used to prevent bigger errors from occurring.
• The configuration of the operating environment of the computer and associated applications and software is explained so that it may be used as outlined in the plan	• Explain and demonstrate how to configure the operating environment of the computer and associated applications and software so that it may be used as outlined in the plan (in Javascript it is not needed since the configuration is automatic and no additional applications and software configuration is needed compared to other scripting languages). • Comparison between traditional software design and validation with W3C. • Conforming to wwwc standards.
• Writing one or more scripts using standard features of scripting languages is explained and demonstrated	• Explain and demonstrate how one or more scripts are written using standard features of a scripting language. JavaScript: control statement Sequence selection Repetition statement

	While Do….while For statement Document. Writeln JavaScript :Arrays JavaScript: functions Events Handlers(onClick,Onload,Onunload,Onsubmit,onreset,onMouseover Onmouseout, onabort,onblur,onchange,onfocus)
• Testing, error identification and correction for most likely conditions in scripts is explained and demonstrated.	• Explain and demonstrate how the scripts are tested, errors identified and corrected for most likely conditions. • Scripting is incorporated in a program like HTML • How multimedia elements are incorporated in a design
ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/Practical work • Assignments • Group work	

SUBJECT OUTCOME	
5.5 Assemble a multimedia, web-based application including scripts	
ASSESSMENT STANDARD	
• The way in which the assembly of a multimedia, web-based application using the saved text, graphics and animation, and written application scripts conforms to the planned specification and user requirements is explained and demonstrated.	• Explain and demonstrate how to assemble a multimedia, web-based application using the saved text, graphics and animation, and written application scripts to conform to the planned specification and user requirements. Practical assembly: • Image • Arrays • Rolling banners • The multimedia source to be retrieved from where it is stored and used in the scripting application. **KEY WORDS, CONCEPTS and APPLICATION** 1) Scripting introduction • What it is • What it is used for • JavaScript and HTML 2) Logic design • Preparing and designing a solution 3) Data types in JavaScript

	• Variables • Working with data types 4) Operators • Building Expressions • Types of operators 5) Functins • Using functions to organize codeEvents 6) Use elements and events • Referencing 7) Control Structures • if…else • if • do…while continue 8) Events Handlers(onClick,Onload,Onunload,Onsubmit,onreset,onMouseover Onmouseout, onabort,onblur,onchange,onfocus) 9) Text and input validation
• The consistency of the content and the function of the application with the design specification and specified	• Explain how the content and function of the application are consistent with the design specification and specified computer

computer system environment are explained.	system environment • Theoretical • Does the script render specification to the design? • Does the webpage render what it is supposed to?
ASSESSMENT TASKS OR ACTIVITIES	
• Class tests • Projects/Practical work • Assignments • Group work	

Progression Table- for Computer Programming Level 2, 3 and 4
Concepts covered in a lower level is assumed to be mastered on a higher level and may thus be included as part of the assessment for the higher level where applicable

Databases (MS Access)			
	Level 2	Level 3	Level 4
		• Create database using multi-tables • Set up relationships between tables • Objects - Tables, Queries, Reports • Create fields in a table - Names (choose relevant names) - Data types – fields such as text, number, date and time, currency, autonumber, Yes/No, memo, , lookup - Properties such as Size / length, Input mask, Default value, Validation rule, Validation text, Required, Text alignment, Decimal places - Choose suitable Primary key • Table manipulation - Sort record - Change field content • Queries - Create basic queries using SQL ▪ SELECT; FROM; WHERE ▪ Boolean operators (and, or, not) ▪ Mathematical operators - 1:M e.g. register class → pupils - Two tables showing master detail relationship with at least one foreign key in one table • Queries from two tables with multiple criteria • Concept of referential integrity • SQL using two tables - SELECT, DISTINCT	

		- WHERE - ORDER BY - GROUP BY - Creating calculated fields - Formatting with round, int, etc. - Create a join query (single joins) using 'WHERE' - Mathematical operators - Special operators: BETWEEN, IN, LIKE - Aggregate function SUM, AVERAGE, MIN, MAX, COUNT - Date functions • Reports - Create reports using a Wizard - Basic editing of reports such as change the heading and add page headers - Display grouped data - Display summary calculations in groups and at end of report (sum, avg, max, min) • Import, Export data	
VB.NET Programming			
	Level 2	Level 3	Level 4
Data structures	• Concept of the way of storing data in a computer so that it can be used efficiently for references to and operations on them • Variables • Simple data types • Converting data types and the effects thereof	• Scope of variables • Data structures - Arrays (1 D) - Arrays (2 D) - Built-in Classes and objects (Dates, Strings, Time, Random) - Structs - Text files	• Scope of variables • Data structures - Classes and objects - Array of objects - Text files - Database – 2 tables
Control structures	Control structures • Sequential • Selection - Single, double and multiple selection • Repetition - Counting	Control structures • Sequential • Selection – nested statements • Repetition – nested statements - Condtional including using a sentinel value (flag)	Control structures as in Level 2, Level 3

	- Conditional		
Operations	• Mathematical operations using mathematical operators – sequence / brackets • Integer arithmetic (dividend and remainder – modulo) • Relational operators • Boolean operators • Simple string manipulation - Length - Position of a character in a string - Obtain a substring from a string - Delete / Insert characters / substrings - Concatenation of strings - Changing case • Validation of input - Valid integer or real number - Data within correct range (e.g. Grade >= 10 AND Grade <= 12)	• String manipulation - Building strings - Obtain a substring from a string - Replace characters in a string (insert and delete) • Operations on and manipulation of 1D arrays - Basic I/O manipulation - Simple sort - Sequential search using a sentinal value (flag) - Simple calculations such as calculating average, sum • Text files: (Sequential) - Linking to a text file - Open, Close - Loop through text file - Import data from text file - Export data to text file • Array of structs - Linking to a text file - Loop through text file into struct objects - Manipulate struct and perform opertations - Import data from text file into the array of structs - Export data to text file from the array of structs - Tracking the number of elements in the arrtay of stucts - Sorting - Processing: sum, average	• Operations on 2 D arrays - Simple operations such as row and column totals • Operations on and manipulation of array of objects. - Tracking the number of elements in an array - Inserting elements - Deleting elements - Sorting - Processing: sum, average • Text files: - Linking to a text file - Open, Close - Loop through text file - Import data from text file - Export data to text file • Classes and objects - Class declaration – attributes and methods - Create object - Instantiation - Encapsulation - Contructor methods - Accessor and mutator methods - Single Inheritance - Composition - Polymorphism • String manipulation - As in Levels 2 & 3 - Using CSV in text files for inputting data
Functions / procedures / methods	• Mathematical functions / procedures / methods - Minimum and maximum values, power,	• User defined functions / procedures / methods using parameters as part of form class including reference parameters	• User defined functions / procedures / methods using parameters as part of class including reference parameters

	square roots, rounding, random values, absolute values, truncation	• Simple user defined functions / procedures / methods using parameters (not var parameters)	
Database and programming			• Access a database through programming language - Setup an OLEDB connection or connect to database by providing path in VB.NET statements - MsAccess Database as designed -> OLEDBConnection [Connectionstring] - DBConnections, DataTables, DBCommands, DataAdapters, DataRows - Create database solution using multi-tables in VB.NET • Data redundancy and Normalisation by inspection • Create a DB Application solution to a set of design spedifications by inserting a programmer designed DB (MDF, data object, designed and created via the object inspector) • Use the DB Wizzards where applicable for the development of a DB Solution • Creating a Multifom DB Application - Develop a DB Solution with a MDI interface - Implement data awere components in the development of the solution e.g. DataGridView • SQL in a VB.NET Application - Display records from a table in a sorted order - Group related data together - Creating calculated fields - Formatting with round, int, etc. - Create a join query (single joins) using 'WHERE' - Mathematical operators

			- Special operators: BETWEEN, IN, LIKE - Aggregate function SUM, AVERAGE, MIN, MAX, COUNT - Date functions - Delete records meeting specified criteria from a table using SQL Statements (DELETE) - Modify records and Update the content of certain fields in a table meeting specified criteria from a table using SQL Statements (UPDATE, SET) - Add records meeting specified criteria from a table using SQL Statements (INSERT) - Implement parameterised queries in the design of the sulution where applicabke • Add data to a database imported from a Sequential textile • Implement user defined classes as part of the DB solution
Error handling	• Syntax • Runtime • Logic • Valid input as indicated in Operations section • Protection blocks - Try and catch - Exception handling	• Protection blocks - Try and catch - Exception handling	• Protection blocks - Try and catch - Exception handling
Debugging	• Trace tables • Use debugger facilities such as watches, traces, breakpoints	• Trace tables • Use debugger facilities such as watches, traces, breakpoints	• Trace tables • Use debugger facilities such as watches, traces, breakpoints
Testing	• Testing for correct output with valid / normal data • Testing for error handling with extreme and errouneous data and invalid responses	• Testing for correct output with valid / normal data • Testing for error handling with extreme and errouneous data and invalid responses	• Testing for correct output with valid / normal data • Testing for error handling with extreme and erroneous data and invalid responses
Documentation	• Comments • Analysis and Design documents (planning	• Comments • Analysis and Design documents (planning	• Comments • Analysis and Design documents (planning

	interface, Input-Processing-Output tables / flowcharts) • User guide	interface, Input-Processing-Output tables / flowcharts) • User guide • Programmer guide (technical guide)	interface, Input-Processing-Output tables / flowcharts, ERD – database) • User guide • Programmer guide (technical guide)
User interface	• Good and bad user interfaces – basic HCI concepts • Appearance • Order • Input - Messages • Organisation of output • Simplicity and clarity	• Error messages (the error message returned should indicate a solution) • Exception handler • Data validation techniques GUI • Appropriate components for input and output • Use of properties with components e.g. tooltips	• Multiform Applications • Navigation between screens • Menus / Page controls • Buttons • Context sensitive help - Projects • Short cuts • Metaphors or images • Consistent behaviour (e.g. F1 for Help • GUI as a front end to an object class where all processing takes place in the object class(es). E.g. an object class, an array of objects class and a front end GUI that instantiates an object of the array of objects.
GUI components	Such as buttons, textfields / areas, labels, panes, checkboxes, radiobuttons, lists, grids, pagecontrol, message dialogues Database:OnPost, OnError, OnBeforeEdit, AfterScroll, etc.		
Events and event handlers	Such as click, mousemove, activate, close, exit, show, create, change, keypress,		
WEB Development			
	Level 2	Level 3	Level 4
HTML			11) Creating a HTML Page • What is XHTML • What is a Web Page • How does a web page work • Elements of a web page • Creating a simple web page - Create a webpage using Notepad - Tags that a web page must have - View webpage in a web browser • Web browsers - Internet Explorer - FireFox

			- Google Chrome - Viewing Web Page Source Code • W3C Validation 2) Elements of a web page • Text formatting and Font Control - Using Boldface and Italics - Font Size - Font Color - Special Characters - Linking to other pages • Headings • Text Areas / Text Fields • Buttons • Radio buttons / groups • Checkboxes • Horizontal Rules • Line Breaks • Adding background color • Lists - Unordered Lists - Ordered Lists - Definition Lists - Nesting Lists 3) Web page design • Forms • Basic Frames • Basic Tables / Nested Tables 4) Web page graphics • Images • Images Maps • Adding background images 5) CSS

Scripting (JavaScript)			1) Scripting introduction • History of scripting • What it is • What it is used for • JavaScript and HTML 2) Logic design • Preparing and designing a solution 3) Data types in JavaScript • Variables • Working with data types 4) Operators • Building Expressions • Types of operators 5) Functions • Using functions to organize codeEvents 6) Use elements and events • Referencing 7) Control Structures • if…else • if • do…while • continue