



**Western Cape
Government**

Education

Western Cape Education Department

E X E M P L A R

JUNE EXAMINATION – 2013

INFORMATION TECHNOLOGY

PAPER 1 – PRACTICAL

TIME: 3 HOURS

MARKS: 120

This paper consists of 10 pages, and there are 6 files in your exam folder.

Make sure that you have a complete set.

INSTRUCTIONS and INFORMATION:

1. THIS IS A THREE-HOUR EXAMINATION. BECAUSE OF THE NATURE OF THIS EXAMINATION IT IS IMPORTANT TO NOTE THAT YOU WILL **NOT** BE ALLOWED TO LEAVE THE EXAMINATION ROOM BEFORE THE END OF THE EXAMINATION SESSION.

2. Log on and open your exam folder. You require the files listed below in order to answer the questions.

QUESTION 1:

Question1.java
DisplayClass.java
JoesGym.mdb

QUESTION 2:

MemberData.txt
GymApp.java

QUESTION 3

RunningTraining.java

3. Save ALL your work in your exam folder under the required names.
NB: Only programs saved in your exam folder will be marked!
4. Save your work regularly as a precaution against power failures.
5. You may make use of the Java's API files. You may NOT consult any other resource material.
6. Type your name and the question number as a comment in the first line of each program, e.g. // Joe Soap, Q. 2.
7. **Attempt ALL questions.**
8. Read the requirements for each program thoroughly and understand the details before starting to program. Do ONLY what is required by each question.
9. In ALL programs marks will be allocated to functional parts, even if the whole program does not work (or hasn't been completed). So make sure that you complete each step before starting the next one.
10. Good luck!

Questions to this paper were contributed by Laurent Kalombo, Zuko Ndabambi and Meyburgh Hofmeyer,
The paper was moderated by Max Brock.

SCENARIO:

Joe's Gym has growing exponentially since its foundation in 2008. To keep up with competition, the manager would like to use the services of a software developer in order for him to be able to handle the day to day operation. You are required to help him with the requests.

Question 1: Joe's Gym – Java Programming and Database

NB: The design of the tables in the “JoesGymDB” database and sample data for this question can be found in Annexure A at the end of your exam paper.

1. Complete the programming code in the **Question1.java** program by creating the necessary SQL statements in the methods as specified below.

1.1 Complete the code in **case A** by formulating an SQL statement to display all the fields in the **Members** table sorted on member's surname and first name. Place the statement in the appropriate line of the program code.

Example of output of the first 7 records:

MemID	MemBDate	MemFName	MemSName	MemCellNr	MemEmail	MemTypeCode	MemFeeOutstanding	MemJoinDate
GOA81185	1980-08-26	Orlando	Adams	0768657208	OAdams@ananzi.co.za	SGM1	100.0	2009-07-18
GTA91094	1974-09-28	Tarren	Adams	0766191158	TAdams@ananzi.co.za	SGM2	0.0	2008-09-29
GRA61151	1971-03-28	Ramiro	Agherdien	0729295712	RAgherdien@hotmail.com	SGM1	100.0	2009-11-01
GDA71079	1971-04-28	Desire	Ah Hing	0848782381	DAH_Hing@Polka.co.za	SGM2	0.0	2009-08-07
GFA51249	1973-04-28	Firdous	Ahmed	0739116468	FAhmed@ananzi.co.za	SGM4	0.0	2009-06-20
GRA11239	1959-04-28	Ra'ees	Ahmed	0728908356	RAhmed@Polka.co.za	SGM4	0.0	2010-10-19
GSA51310	1987-10-28	Shiraz	Ajam	0722853184	SAjam@ananzi.co.za	SGM4	0.0	2008-09-05

(4)

1.2 Joe's Gym is planning on hosting a special event for all members who have their birthday in the next month (June). In order to phone these members and invite them personally, the receptionist needs a list of names and phone numbers. The manager also needs the list with ages and birthdates sorted by the day. Complete the code in **case B** by formulating an SQL statement to display these members' *first name, surname, cell number, age on the day (a calculated field called Age)* and the *date of birth* sorted by the day. Place the statement in the appropriate line of the program code.

Example of output of the first 7 records:

MemFName	MemSName	MemCellNr	Age	MemBDate
Bernita	Camons	0844831063	65	1948-06-02
Shaun	Scheinberg	0761316352	57	1956-06-04
Shane	Renshaw	0768894661	29	1984-06-04
Roxanne	Kimfley	0728102858	41	1972-06-08
Mandisa	Mapela	0768431086	41	1972-06-11
Daryn	Songwiqui	0769305602	41	1972-06-13
Rhys	Mawela	0733468985	37	1976-06-18

(7)

1.3 The manager wants to launch a new promotion but believes that people who have gmail addresses are poor and will never be able to afford the new promotion. For marketing purposes he would therefore like to have a list of all members who do **NOT** have their email with gmail. Complete the code in **case C** by formulating an SQL statement to display the *name and surname, email and contact number* fields for all these members from the **Members** table to generate a list to give him. Place the statement in the appropriate line of the program code.

Example of output of the first 7 records:

MemEmail	MemFName	MemSName	MemCellNr
ABadela@tut.ac.za	Akhona	Badela	0847057127
ABrett@Polka.co.za	Ashleigh	Brett	0832451108
ABili@yahoo.co.za	Abigail	Bili	0841724360
ABlaaw@yahoo.co.za	Aileen	Blaaw	0845499992
ABreidenbach@tut.ac.za	Andrea	Breidenbach	0846548092
ABritz@tut.ac.za	Ashley	Britz	0766242309
ABowren@yahoo.co.za	Amy	Bowren	0762293049

(5)

- 1.4 The manager has just realised that some the members of type Basic Orange have not paid their fees and wants to know how many members are involved. Complete the code in **case D** by formulating an SQL statement that will count the number of members with membership code SGM1 (= Basic Orange) and outstanding fees. Call this number “*NumMembersWithFeesOutstanding*” Place the statement in the appropriate line of the program code in **case D**.

```
NumMembersWithFeesOutstanding
=====
                                20
```

(5)

- 1.5 From time to time members cancel their membership at Joe’s Gym and need to be removed from the database.

Complete the code in **case E** by formulating an SQL statement to use the membership code read in from keyboard in order to remove that member. Use the method to remove member GZX81326 (the second last member). Use method A to check whether that member has in fact been removed. Example of output:

```
Please enter the membership code:
GZX81326
Records Processed Successfully
```

(4)

- 1.6 Some members have approached the gym requesting to view their profile. Complete the code for the **Option F** menu option by asking the user to enter the **Membership ID** . Formulate an SQL statement to display the *MemID*, *MemFName*, *MemSAID*, *MemTMFee*, *MTypeAccLevel* of the member. **NB:** The fields are from two different tables; so the **Members** table and **MShipTypes** table must be connected on the primary key.

```
Please enter the membership code:
GZX81326

MemID      MemFName  MemSName  MemTMFee
=====
GZX81326   Zethu       Xotyeni   180.0
```

(6)

- 1.7 The manager would like to be able to record new members in the **Members** table in the database.

Complete the code in **case G** by formulating an SQL statement to accomplish this. Place the statement in the appropriate line of the program code.

Test your SQL statement with the following data:

```
Member id: GST20185
Member birth date: 31/05/1953
Member first name: Sibusiso
Member surname: Thembani
Member Cell : 0768562655
Member e-mail: SThembani@gmail.com
Member type code: SGM3
Member fee outstanding 30: 0
Member join Date: 2013/05/30
```

```
Records Processed Successfully
```

(7)

Total Question 1 = [38]

Question 2: Java – Object-oriented Programming

2.1 Start a new Java class named **GymMember.java**. (1)

2.1.1 Add the following private fields to this class:

fName – the first name of the member

sName – the surname of the member

idNum – the identification number of the member

feesOwing – the fees that the member owes the gym

Ensure that you choose an appropriate data type for each of these fields. Note that the fees MUST be stored in a double! (1)

2.1.2 Write a parameterised constructor for all 4 attributes. The parameters should be used to initialise the fields of the class. (2)

2.1.3 Write the following accessor methods that will return the values as indicated:

2.1.3.1 The **getfName()** method that will return the first name of the member.

2.1.3.2 The **getsName()** method that will return the surname of the member.

2.1.3.3 The **getFeesOwing()** method that will return the amount that the member owes the gym. (3)

2.1.4 Write a **getGender ()** method. It should NOT set the value of an attribute (there is no 'gender' attribute) but return the gender (just **m** or **f**) of the member based on his/her ID number.

The first 6 digits of an ID number are for the date of birth. **The next digit after that indicates the gender:** if it is between 0 and 4 the person is female, if it is between 5 and 9 the person is male. (5)

2.1.5 Write a **toString()** method that displays the basic information on each member arranged as follows (here with data from the first record):

First Name = Alicia

Surname = August

ID Num = 5304281549084

Gender = f

Fees Owing = R0.00

<empty line>

The gender does not refer to an attribute but must call the method written in 2.1.4.
The fees owing must have an 'R' in front and must be rounded off to two decimal places. (6)

Subtotal 2.1 [18]

2.2 You have been provided with an application class named **GymApp.java**.

2.2.1 Create an array named **memberArr** that can hold a maximum of 1000 **GymMember** objects. (2)

2.2.2 Write code that reads the information from the **MemberData.txt** file into the array you created in 2.1 bearing the following in mind:

- Read in a line of text from the text file. (5)
- Separate the text into the first name, surname, id number and the outstanding fees. (3)
- Use the items of data to create a new **GymMember** object and place it into the object array you created in 2.2.1. (3)
- Use a counter to keep track of how many objects are placed in the array. (2) (13)

2.2.3 Write code to complete the options on the menu provided in the **GymApp.java** file as indicated below. The methods from the **GymMember.java** file should be used wherever applicable.

- (a) **Members List:** Display a suitable heading as illustrated here. Display a list of all the members.

```
MEMBERS
=====
First Name = Alicia
Surname = August
ID Num = 5304281549084
Gender = f
Fees Owing = R0.00
```

```
First Name = Akhona
Surname = Badela
ID Num = 5510281986084
Gender = f
Fees Owing = R180.00
```

(3)

- (b) **Alphabetic selection:**

Allow the user to enter a letter of the alphabet from the keyboard. The program must then list all the names of the members whose surnames start with the letter chosen.

The format of the name must be <surname><comma><space><firstName>

```
Alphabetical selection:
=====
Enter a letter of the alphabet: x
Xotyeni, Masixole
Xotyeni, Zethu
```

(8)

- (c) **Fees outstanding:**

The program must go through the memberArray and add up what all the members that have outstanding fees owe. At the same time the program must count how many members owe money to the gym and then work out the average just for members who are not paid-up.

The fees owing and the average must have an 'R' in front and must be rounded off to two decimal places. (11)

77 members owe a total of R17030.00 at an average of R221.17 each.

Subtotal 2.2 [37]

Total Question 2 = [55]

Question 3: Problem-Solving

Sibu is training for the Two Oceans Half Marathon in 2014. He has been training for a year and his total monthly distances have been recorded in an array called **distances**. You have also been supplied with an array called **months** that contains the corresponding months' names.

Kilometres trained on the road and on a treadmill in the gym:

Months	July	August	September	October	November	December	January	February	March	April	May	June
Distances (km)	90	100	103	145	77	135	90	107	99	128	23	117

You are required to complete the following three methods.

Open the **RunningTraining.java** class and edit the methods in the class so that all the methods execute correctly when the corresponding menu option is selected.

3.1 Menu Option A

Complete the displayLog() method so that the contents of both arrays will be displayed as follows:

Month	Distance
July	90
August	100
September	103
October	145
November	77
December	135
January	90
February	107
March	99
April	128
May	23
June	177

(4)

3.2 Menu Option B

Sibu needs to be able to edit the total distance trained in any month. Complete the code for the editLog() method as follows:

Ask the user for the month to be changed. The user must type in the full month name (correct spelling counts but not whether upper or lower case is used) and the new distance for that month. Use this data to update the distance array.

Your program must display an error message if the month name has not been spelt correctly (but not if e.g. it hasn't been started with a capital letter).

Hint: Test your interface by updating *May* to 111km and then choose the menu option A to view your update. (12)

```
Please enter month you want to update: May
Please enter the new number of kilometres: 111
May updated
```

3.3 Menu Option C

Sibu would like to have a report displayed on screen. Complete the logReport() method to display the following information:

3.3.1 Sibus target is 120 km per month. Display the month name and “achieved” next to it, for all the months that he ran 120km or more.

3.3.2 Calculate and display the average monthly distance that Sibu had logged over the year of training.

Sample Report Output

```
Sibu's achievement of his running goal of 120km per month
July
August
September
October      achieved
November
December     achieved
January
February
March
April        achieved
May
June         achieved
```

```
Average monthly distance: 114.5 km
```

(11)

Total Question 3 = [27]**GRAND TOTAL = 120**

ANNEXURE A: Table Description Sheet for Database

This sheet shows the data structures and sample data for the tables used in the JoesGymDB database for Question1.

Members – Table Structure:

Field name	Type and size	Description
MemID	Text, 8	The member's membership code, the primary key
MemBDate	Date/Time	The member's birth date
MemFName	Text, 20	The member's first name
MemSName	Text, 20	The member's surname
MemCellNr	Text, 10	The member's cell phone number
MemEmail	Text, 30	The member's email address
MemTypeCode	Text, 4	The code of the type of membership
MemFeeOutstanding	Number, Double	The amount (of the fees) the member is owing
MemJoinDate	Date/Time	The date the member joined the gym

MShipTypes – Table Structure:

Field name	Type and size	Description
MemTypeID	Text, 4	The code of the membership type, the primary key
MemTName	Text, 20	The name of the membership type
MemTMFee	Number, Double	The annual fee for the type of membership
MTypeAccLevel	Number, Integer	The access level for the type of gym membership

Members – Data Sample

MShipTypes – Data

Members									MShipTypes			
MemID	MemBDate	MemFName	MemSName	MemCellNr	MemEmail	MemTypeCode	MemFeeOutstar	MemJoinDate	MemTypeID	MemTName	MemTMFee	MTypeAccLe
GAA41215	1953/04/28	Alicia	August	0844949538	AAugust@gmail.com	SGM4	0	2010/11/1	SGM1	Basic Orange	100	2
GAB11471	1955/10/28	Akhona	Badela	0847057127	ABadela@tut.ac.za	SGM2	180	2009/08/2	SGM2	Extended Silver	180	3
GAB31086	1977/06/28	Ashleigh	Brett	0832451108	ABrett@Polka.co.za	SGM1	0	2008/10/1	SGM3	Premium Gold	280	4
GAB31498	1954/10/28	Abigail	Bili	0841724360	ABili@yahoo.co.za	SGM3	280	2009/10/0	SGM4	Platinum Select	350	5
GAB51158	1956/06/23	Aileen	Blaaw	0845499992	ABlaaw@yahoo.co.za	SGM2	0	2010/03/2				
GAB51364	1957/04/28	Andrea	Breidenbach	0846548092	ABreidenbach@tut.ac.za	SGM3	0	2008/10/10				
GAB51473	1955/03/28	Ashley	Britz	0766242309	ABritz@tut.ac.za	SGM2	0	2010/05/06				
GAB71438	1987/08/28	Aimee	Bobie	0724692618	ABobie@gmail.com	SGM1	0	2008/11/30				
GAB81177	1966/03/28	Alexa	Botha	0725816905	ABotha@gmail.com	SGM2	0	2008/03/27				
GAB91298	1969/10/28	Amy	Bowren	0762293049	ABowren@yahoo.co.za	SGM2	0	2008/03/02				
GAC31154	1969/05/28	Anjli	Camagu	0729343067	ACamagu@Hotmail.com	SGM1	100	2010/01/31				
GAC31388	1967/04/28	Asanda	Cata	0838831403	ACata@gmail.com	SGM2	0	2010/11/10				
GAD21441	1974/08/28	Anton	Dorfling	0823268196	ADorfling@gmail.com	SGM3	0	2008/10/27				
GAD71077	1972/03/09	Ashanti	Daya	0838057759	ADaya@gmail.com	SGM4	350	2009/06/16				
GAF91240	1977/08/28	Aspasia	Fudu	0824247937	AFudu@gmail.com	SGM4	350	2010/11/09				
GAG61138	1972/07/02	Avela	Goosen	0844704916	AGoosen@ananzi.co.za	SGM3	0	2008/06/28				
GAG71338	1972/07/05	Ashleigh	Goliath	0847594532	AGoliath@yahoo.co.za	SGM1	0	2008/08/01				