



eduresourceza.com



DATE: JUNE 2011 EXAMINER: W du Toit, D Kench MODERATOR: L Coker	Information Technology GRADE 11 Practical	Examination TIME: 3 hours and a half hour for printing TOTAL: 120
--	--	--

LOs Targeted: LO4: Programming and Software Development	ASs Targeted: 11.4.1, 11.4.4, 11.4.11, 11.4.12, 11.4.13, 11.4.18, 11.4.19
--	---

READ THE FOLLOWING INSTRUCTIONS

1. This paper consists of 4 pages. Check that your paper is complete.
2. This question paper is to be answered using Object Oriented Programming principles. Your program must make sensible use of methods and parameters.
3. Only answer the question that is stated. For example, if the question does not ask for data validation, then no marks are awarded for it, and therefore no code needs to be written.
4. If you cannot get a section of code to work, comment it out so that it will not be executed and so that you can continue with the examination.
5. You must save all your work regularly on the disk you have been given, or on the disk space allocated to you for this examination
6. If there is a technical interruption that prevents you from writing your examination, such as a power failure, when you resume writing your examinations, you will only be given the time that was remaining when the interruption began. No extra time will be given to catch up work that was not saved.
7. Print a code listing of all the programs/ classes that you code. Print the output from your program, if possible. You will be given half an hour to print after the examination is finished.

SECTION A – OBJECT ORIENTED PROGRAMMING**[80 MARKS]****Scenario**

We obtained some prices from www.adventurebookings.co.za about extreme sports activities across the country. This information can be used for team building in companies as well as for special birthday gifts.

The files required for this question will be found on your T: drive

A list of the extreme sports, where it takes place and the cost is stored in a text file called **"SportFile.txt"**. The contents of this file is as follows:

```
Abseiling#Table Mountain#595
Whale Watching#Pletternberg Bay#500
Whale Watching#Mosselbay #550
Sky Diving#Plettenberg Bay#1800
Sky Diving#Pietermaritzburg#1650
Sky Diving#Mosselbay#2000
Paragliding#Enkhaba Mountain#875
Paragliding#Lions Head#950
River Rafting#Parys#240
Bungee Jumping#Bloukrans#690
Bungee Jumping#Orlando owers#480
Hot Air Balloon#Paarl#2850
```

Each line is divided into 3 fields, each separated by a #. The fields are in the order of **sportname**, **place** where it takes place and the **cost** of the activity.

Question 1:

Create a new class called **Sport**.

- 1.1. Create fields with appropriate types to store the **name** of the sport, the **place** it takes place and the **cost**. Also declare an extra field of type String, for a **code** that you are going to generate later. (4)
- 1.2. Code a parameterised **constructor** method. The first 3 fields get their values from the parameters. The code will get its value by calling the method **generateCode** in question 1.5. (4)
- 1.3. Code a **toString** method that will return the fields of this class and display them using tabs to separate the fields. (4)
- 1.4. Code an accessor method as well as a mutator method for the cost field. (4)
- 1.5. Code a **generateCode** method that will determine the code for each extreme sport as follows:

The first letter of the code will indicate whether it is a

- water sport (**W**)
- an air sport (**A**)
- a land sport (**L**).

In the given data the *Whale Watching* and *River Rafting* are water activities (W), the *Bungee Jumping* and *Abseiling* are land activities (L) and the *Sky Diving*, *Paragliding* and *Hot Air Balloon* are air activities (A).

The second part of the code is determined on the cost of the activity. This should indicate how many 100s are in the cost.

For example:

Hot Air Balloon at R540 : Code = A5 (Air activity, 5 hundreds in cost)

Whale Watching at R900: Code = W9 (Water activity, 9 hundreds in cost) (8)

[24 Marks]

Question 2:

Code a class called **SportArr** to store the items of all the different activities.

2.1. Create two instance variables for this class:

- An array that can store up to 20 objects of TSport and
- A variable to keep track of the number of elements in the array. (4)

2.2. Code an appropriate **constructor** method that will read all the data from the text file "*SportFile.txt*" and instantiate the array of sport activities, using the array. (8)

Pay attention to the following:

- Include exception handling that will determine whether the file can be found and display an appropriate message. (4)
- Check if the cost is a positive. If it is negative then make the number positive. (5)

2.3. Code a **toString** method. This method returns the details of all the sport activities, each on a new line. (5)

2.4. Code a method **averageCost**. This method must loop through the array in order to determine the total cost of all the sport activities and then determine the average cost per activity. Return the answer as part of a sentence: "The average cost is R - - -" (6)

2.5. Code a method called **sortByCost**. This method must sort the data in and find the 5 cheapest items in the array. (8)

2.6. Code a method **changePrice** that will increase the cost of all the sky diving entries by R500. . This method must search the array and then use the **setCost** method in the Sport class. (8)

[48 Marks]

Question 3:

Now add the following code in your application class called **SportsApp** to

- 3.1. Instantiate a SportArr object.
- 3.2. Display all the sports (question 2.3)
- 3.3 Display the average cost (question 2.4)
- 3.4 Display the 5 cheapest sports. (question 2.5)
- 3.5 Change the price of the sky diving and then redisplay the array. (question 2.6 and question (2.3)

[8]**SECTION A – STRUCTURED QUERY LANGUAGE****[40 MARKS]**

An MS Access database file called Animals is given on your T: drive. Migrate the database to MySQL and connect to the database using Netbeans.

Perform the SQL statements using your database and copy your answers to the ANSWER file that is also on your T: drive. Make sure you put your name in the file.

A zoo has created a database about its animals and some of their properties. The details are stored in a table called tblAnimals in the Animal database. A description of the fields in the table tblAnimals are given below:

```
CREATE TABLE `tblanimals` (
  `Name` varchar(60) DEFAULT NULL,
  `Class` varchar(60) DEFAULT NULL,
  `Location` varchar(60) DEFAULT NULL,
  `Habitat` varchar(60) DEFAULT NULL,
  `Weight (kg)` double(7,2) DEFAULT NULL,
  `Length (cm)` double(7,2) DEFAULT NULL,
  `Diet` varchar(60) DEFAULT NULL,
  `Covering` varchar(60) DEFAULT NULL,
  `Life Span (years)` double(7,2) DEFAULT NULL
)
```

A sample of the data is given below showing the first few rows.

Name	Class	Location	Habitat	Weight (kg)	Length (cm)	Diet	Covering	Life Span (years)
Antelope	Mammalia	Africa	Savannah, grassland	280	240	Plant eater	Fur	17
Chimpanzee	Mammalia	Africa	Rainforest	89	120	Omnivore	Fur	40
Python	Reptilia	Africa	Savannah, grassland	45	650	Carnivore	Scales	28
Rhinoceros	Mammalia	Africa	Savannah, grassland	1360	350	Plant eater	Thick skin	40
Anteater	Mammalia	South	Rainforest, savannah, grassland	39	200	Insectivore	Fur	9
Giant Panda	Mammalia	Asia	Forest, rainforest, grassland, mountain	160	170	Omnivore	Fur	15
Polar Bear	Mammalia	North	Tundra	650	300	Carnivore	Fur	20
Kangaroo	Mammalia	Australia	Grassland	65	200	Plant eater	Fur	6
Mountain Lion	Mammalia	North	Mountain, grassland, forest	103	243	Carnivore	Fur	12
Penguin	Aves	Antarctica	Tundra	4	112	Carnivore	Feathers	20
Parrot	Aves	South	Rainforest	1	100	Omnivore	Feathers	75
Eagle	Aves	North	Forest, grassland, desert, mountain	5	100	Carnivore	Feathers	30

Perform the following SQL statements

1. Sort all the Animals according to Class then Name alphabetically. (4)
 2. Generate a code for each animal using the first two letters of their name and their life span. (5)
 3. Count the number of animals that have fur as a covering. (5)
 4. Find all the omnivores found in Africa or Asia. (6)
 5. Change all the lengths from centimetres to meters as using the field LengthMetres. (4)
 6. Find the top 10 animals with the shortest life span. (5)
 7. Find the average weight of each class of animal. (5)
 8. Change the Eagle to weigh 6 kg. (3)
 9. Delete any animal that does not have a covering. (4)
-

TOTAL : 120