



eduresourceza.com



<u>DATE:</u> NOV 2012 <u>EXAMINER:</u> D Kench	Information Technology GRADE 11 Practical	Examination <u>TIME:</u> 3 hours and a half hour for printing <u>TOTAL:</u> 120
---	--	---

Name

User Name

Password

READ THE FOLLOWING INSTRUCTIONS

1. This paper consists of **5** 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 – STRUCTURED QUERY LANGUAGE**[40 MARKS]**

An MS Access database file called **TollGate** 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.

The company called IT Dreams has been commissioned to provide an IT solution for the new Toll Gates. A toll gate is placed on a highway and any car that passes under the toll gate is automatically recorded. The registration, the name of the toll gate and date time stamp of when the car passed under the toll gate is stored in a table in the **TollGate** database. The table's name is **tblTollGate**. A second table called **tblCars** stores the details of the car: the registration number, the make, model and colour. The company has populated the database with some sample data to check if the design of the database is correct.

Design View of tblCars

tblCars	
Field Name	Data Type
RegNo	Text
Make	Text
Model	Text
Colour	Text

Sample Data of tblCars

tblCars			
RegNo	Make	Model	Colour
BM 78 GT GP	Kia	Carnival	Red
BMP 456 GP	Honda	Jazz	Blue
GMP 657 GP	Nissan	GTR	Black
HPT 546 GP	Honda	Civic	Blue
Kassie	Kia	Seronto	Green
NKP 042 GP	Toyota	Corolla	White
PM 45 HK GP	Mercedes	SLK200	Blue
Six Pack	Ford	Fiesta	Green
Toy4U2C	Toyota	Fortuner	Silver
TR 76 KP GP	Nissan	Micro	White
WPT 324 GP	Toyoto	Yaris	Black
ZMY 564 GP	Mercedes	C Class	White

Design View of tblTollGate

tblTollGate	
Field Name	Data Type
RegNo	Text
TollGate	Text
DateTime	Date/Time

Sample Data of **tblTollGate**

RegNo	TollGate	DateTime
BM 78 GT GP	Heron	2012/11/02 06:08:12 AM
Six Pack	Heron	2012/11/02 07:09:20 AM
GMP 657 GP	Stork	2012/11/02 07:45:30 AM
ZMY 564 GP	Heron	2012/11/02 08:20:45 AM
WPT 324 GP	Heron	2012/11/02 08:30:15 AM
WPT 324 GP	Stork	2012/11/02 04:00:23 PM
ZMY 564 GP	Heron	2012/11/02 05:23:32 PM
GMP 657 GP	Stork	2012/11/02 06:15:24 PM
BM 78 GT GP	Heron	2012/11/02 08:18:22 PM
Kassie	Stork	2012/11/02 11:09:05 PM
Six Pack	Heron	2012/11/03 07:10:50 AM
GMP 657 GP	Stork	2012/11/03 07:41:13 AM
Toy4U2C	Heron	2012/11/03 02:10:37 PM
Kassie	Heron	2012/11/03 03:51:56 PM
BM 78 GT GP	Heron	2012/11/03 06:15:28 PM
GMP 657 GP	Stork	2012/11/03 07:45:30 PM
BM 78 GT GP	Heron	2012/11/03 09:01:47 PM

Perform the following SQL statements

- Sort all the cars by make then by model. (4)
- List all the registration numbers that have passed through the Heron tollgate. Display only the registration number. (5)
- Count the number of Toyotas. Name this field **ToyotaCount**. (5)
- Count the number of cars that have passed through each toll gate. (5)
- List all the cars that have passed through the Heron tollgate before 11:00 AM on the 2 November. (5)
- Find the first three cars that have passed through the Stork tollgate. (5)
- Insert a new car. Use any registration number, make, model and colour. (3)
- Delete the car with the Registration 'Kassie'. (3)
- Create a code for each car. Use the first 2 letters of the registration number, the third letter of the make and the last 2 letters of the model. (5)

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

A text file **TollGate.txt** has been generated containing the details of each car and the number of times the car has passed through the toll gate Heron. Each field in the file is separated by a comma (,):

Registration number, Make, Model, Colour, Number of Access through Toll Gate Heron

```
BM 78 GT GP,Kia,Carnival,Red,3
BMP 456 GP,Honda,Jazz,Blue,4
GMP 657 GP,Nissan,GTR,Black,5
HPT 546 GP,Honda,Civic,Blue,1
Kassie,Kia,Seronto,Green,8
NKP 042 GP,Toyota,Corolla,White,20
PM 45 HK GP,Mercedes,SLK200,Blue,3
Six Pack,Ford,Fiesta,Green,6
Toy4U2C,Toyota,Fortuner,Silver,2
TR 7 KP GP,Nissan,Micro,White,3
WPT 324 GP,Toyoto,Yaris,Black,8
ZMY 564 GP,Mercedes,C Class,White,9
```

Question 1:

Create a new class called **Car**.

- 1.1. Create fields with appropriate types to store the car registration, make, model, colour and number of accesses. Include a Boolean field called **validReg** that will store a true if the registration of the car is valid and false if the registration is not valid or cannot be checked. (4)
- 1.2. Code a parameterised **constructor** method to assign values to the registration number, make, model, colour and number of accesses. (4)
- 1.3. Code a **toString** method that will return ALL the fields of this class and display them using tabs to separate the fields. (4)
- 1.4. Code accessor methods for the registration number, number of access and validReg fields. (3)
- 1.5. Code a mutator method for the number of accesses field. (1)
- 1.5. Code a **checkReg** method that will determine if the registration number is valid. A registration number is valid if it is either:
 - three letters (excluding vowels), three numbers and then “GP” separated by one space.
 - two letters (excluding vowels), two numbers, two letters (excluding vowels) and then “GP” separated by one space.
 - if the registration is a set of words then the registration is considered to be not valid and **validReg** must be set to false. *(In reality this would be checked against*

The **checkReg** method must be called from the constructor and must be used to assign a Boolean value to the **validReg** field. (15)

[31 Marks]

Question 2:

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

- 2.1. Create two instance variables for this class:
 - (a) An array that can store up to 200 objects of Cars and
 - (b) 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 **TollGate.txt** and instantiate the array of cars, using the array. (8)
- 2.3. Code a **toString** method. This method returns the details of all the cars, each on a new line. (5)
- 2.4. Code a method called **invalidReg** to display all the details of the car's whose registration number is considered to be invalid. (6)
- 2.5. Code a method called **sort** to sort the cars according to registration number. (6)
- 2.4. Code a method called **search** to search for a car given the car's registration number. Return the **Car** object or null if the car cannot be found (6)
- 2.5. Code a method called **updateCars** that will update the number of times each car has passes through the toll gate. A text file called **Passes.txt** stores the registration number of each car that has passed through the toll gate. Some cars have passed through more than once and hence their registration number is listed multiple times in the text file. The method must update the details of each car with each and every occurrence in the text file. For example the car with the registration Kassie is listed 3 times. The car has already has a total of 8 accesses and will produce a new total of 11. The number of accesses field for each car must be updated with the new value.

A sample of the file **Passes.txt** is shown below:

```
HPT 546 GP
Kassie
BM 78 GT GP
Six Pack
Six Pack
NKP 042 GP
Toy4U2C
BMP 456 GP
Kassie
BMP 456 GP
Toy4U2C
NKP 042 GP
GMP 657 GP
HPT 546 GP
Kassie
NKP 042 GP
PM 45 HK GP
Toy4U2C
```

(8)

[43 Marks]

Question 3:

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

- 3.1. Instantiate a CarArr object. (1)
- 3.2. Display all the cars sorted alphabetically. (2)
- 3.3. Display the cars with invalid registrations. (1)
- 3.4. Update the accesses and display all the cars. (2)
- 3.5. Display all the cars with the updated accesses.

[6 Marks]

TOTAL : 120