



eduresourceza.com

PRELIM 2018
Information Technology
Paper 2 Practical

TIME: 3 HOURS

MARKS: 120

EXAMINER: CLUSTER

MODERATOR: CLUSTER

This question paper consists of 10 pages (cover page included)

SURNAME: «Surname»

USER NAME: «sAMAccountName»

PASSWORD: «Password»

PLEASE READ THE FOLLOWING INSTRUCTIONS CAREFULLY

1. Log into the exam profile allocated to you.
2. This question paper is to be answered using Object-Oriented Programming principles. Your program must make sensible use of methods and parameters.
3. This paper is divided into two sections. All candidates must answer both sections.
4. This paper is set in programming terms that are not specific to any particular programming language (Delphi/Java) or database (Access/MySQL).
5. Make sure that you answer the questions in the manner described because marks will be awarded for your solution according to the specifications that are given in the question.
6. Only answer what is asked in each question. 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.
7. 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. If possible, try to explain the error to aid the marker.
8. Your programs must be coded in such a way that they will work with any data and not just the sample data supplied or any data extracts that appear in the question paper. You are advised to look at the supplied data files carefully.
9. Make sure that routines such as searches, sorts and selections are to be developed from first principles, and that you do not use the built-in features of a programming language for any of these routines.
10. All data structures must be defined and declared by you, the programmer. You may not use components provided within the interface to store and later retrieve data.
11. Read the whole question paper before you choose a data structure. You may find that there could be an alternative method of representing the data which will be more efficient considering the questions that are asked in the paper.
12. You must save all your work regularly on the disk you have been given, or the disk space allocated to you for this examination.
13. If there is a technical interruption that prevents you from writing your examination, e.g. 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.
14. Make sure that your name appears as a comment in every unit that you code as well as on every page of hard copy that you hand in.
15. Print a code listing of all the programs/classes that you code. Printing must be done after the examination. You will be given half an hour to print after the examination is finished. Your teacher will tell you what arrangements have been made for the printing of your work.

SCENARIO

We live in a Cashless Society where paying with a bank card or preloaded cash card is the norm of the day. Children are taught from a young age to use cards when school tuckshops use preloaded card payments.

Secure online transactions have become a popular way of shopping, the client simply provides their card details and the transaction is made. These online shopping sites encourage customers to create wish lists. This is a great marketing tool for companies to send customised advertising to customers based on the contents of their wishes.

SECTION A**SQL****[40]****Question 1**

Many schools are making use of a cashless tuckshop system. Students are given a prepaid lunch card which they can use to make purchases at the tuckshop. Because the tuckshop can only prepare limited quantities of food, students can only buy ONE menu item per day.

The given database called **CashlessTuckshop** contains three tables: **tblStudents**, **tblSales** and **tblMenu**. The data from the tables are detailed below with screenshots of a selection of the data contained in each table.

tblStudents		
Field Name	Data Type	Description
CardNumber	Number	This assigns a unique card number to each student
StudentName	Text	This field contains the name of the student
Grade	Number	This field contains the grade of the student
DateOfBirth	Date/Time	This field contains the students date of birth
CardLimit	Currency	This field contains the spending limit of each student
Allergy	Text	This field contains the student's food allergy

tblStudents					
CardNumber	StudentName	Grade	DateOfBirth	CardLimit	Allergy
100	Ali Rodriguez	11	2000/12/24	R 500.00	Sugar
102	Isaiah Garza	10	2001/01/20	R 200.00	Gluten
103	Tatiana Mckenzie	9	2002/06/15	R 200.00	
105	Claudia Whitney	8	2003/05/25	R 200.00	Gluten
106	Fleur Howell	10	2001/10/24	R 200.00	Sugar
107	Damon Hampton	10	2001/09/14	R 400.00	Nuts
110	Branden Lynn	8	2003/05/16	R 200.00	

tblSales		
Field Name	Data Type	Description
SaleNumber	AutoNumber	This assigns a unique number to each sale
DateOfSale	Date/Time	This field contains the date that the sale occurred
CNumber	Number	This field contains the number of the card that the sale was charged to
Quantity	Number	This field contains quantity of the item
ICode	Text	This field contains the item code of the food that was bought

tblSales			
SaleNumber	DateOfSale	CNumber	INumber
1	2017-02-01	610	BUR1
2	2017-02-01	147	BUR1
3	2017-02-01	299	BUR3
4	2017-02-01	376	SAL2
5	2017-02-01	158	BUR1
6	2017-02-02	388	SAL4
7	2017-02-03	148	BUR4
8	2017-02-03	324	ADM1
9	2017-02-03	297	BUR4
10	2017-02-04	474	BUR3

tblMenu		
Field Name	Data Type	Description
ItemCode	Text	This assigns a unique number to product
ItemCategory	Text	This field contains the category into which each product falls.
ItemDescription	Text	This field contains a description of the product
Cost	Currency	This field contains the amount that each product costs
Allergen	Text	This field contains the allergen that the food can have
StockAvailable	Number	This is the amount of stock for each product

tblMenu					
ItemCode	ItemCategory	ItemDescription	Cost	Allergen	StockAvailable
ADM1	Misc Menu	Sour Dough toast with home made jams	R 10.00	Sugar	54
ADM2	Misc Menu	Toasted Almond Muesli	R 25.00	Nuts	37
ADM3	Misc Menu	Three Cheese Toastie on Rye	R 20.00	Diary	89
ADM4	Misc Menu	Avo on Toast	R 20.00		52
BUR1	Burger	Cheese Burger	R 40.00	Gluten	44
BUR2	Burger	Beef n Bacon Burger	R 45.00	Gluten	79
BUR3	Burger	Panko Buttermilk Chicken Gluten Free Burger	R 40.00	Diary	32
BUR4	Burger	Vegan Bahn, mushroom pattie, gluten free	R 50.00		29
SAL1	Salad & Things	Old Bay Season Chips	R 20.00		37
SAL2	Salad & Things	Apple and Strawberry Salad	R 40.00		61
SAL3	Salad & Things	Chicken Ceasar Salad	R 35.00		42
SAL4	Salad & Things	Cous Cous and Orange Salad	R 35.00		58

SECTION A

[40]

Write SQL statements to help the tuckshop manager Mrs Smith with the following queries:

- 1.1 She needs a pricelist that she can put up in the tuckshop. It must contain the details of all the menu items sorted alphabetically by category and then description. (3)
- 1.2 Display all the details of the most expensive item on the menu. (2)
- 1.3 The grade 8 card limits must be increased by 15%. (3)
- 1.4 Only students older than 15 are allowed to buy the energy drink (BEV1). Display a list with the names of students who are 15 or younger. (4)
- 1.5 Mrs Smith more wants to ensure that a student does not buy a menu items with ingredients to which they are allergic. Display a list with the student name, grade, allergy and menu item that contains the allergen the student is allergic to sorted by student name. (5)
- 1.6 Mrs Smith needs to inform each student at the end of the month how many purchases they made and how much money they have left on their card. Display a list with the cardNumber, student name, number of transactions and amount left on the card to spend. An example of the output is shown below: (7)

Name	NumTransactions	Remain
Aaron Tillman	5	R55.0000
Abbot Woodward	1	R165.0000
Abigail Clark	1	R180.0000
Alan Hall	1	R355.0000
Alexa Castro	4	R240.0000
Alexis Carey	1	R380.0000
Alfreda Branch	1	R160.0000
Ali Rodriguez	1	R480.0000
Ann Harding	1	R255.0000
April Daniels	4	R235.0000

- 1.7 Insert a new menu item Hawaiian Burger in the table called **tblMenu**. Use the **itemcode** BUR5. The tuckshop will make 50 daily. It contains gluten as an allergen. The cost of this item is the average cost of all the burgers. (6)
- 1.8 Mrs Smith needs to know which students did not buy anything from the tuck shop. Display a list with the student name and the card limit of all the students who has not yet used their card at the tuck shop. (4)

- 1.9 The local gym will award prizes to students with healthy eating habits. Display a list with the card numbers of all students who bought any salad menu item. The card number must only be displayed once. Include a random number between 1 and 20 call this field **PrizeCode**. The gym will use this code to determine the prize that each healthy student will receive. (6)

CardNumber	PrizeCode
376	14.0
388	4.0
110	20.0
244	2.0
222	13.0
338	15.0
333	18.0
170	20.0
113	4.0
616	2.0

SECTION B

OBJECT-ORIENTED PROGRAMMING

[80]

BuyAllYouCan is a website that allows users to purchase items over the internet. The website carries items of three categories, i.e. Computer, Photography and Security. A user can browse the website and create a wish list and purchase items. A user logs in and can add or remove any number of items to their wish list.

The user has the following details:

- **LoginId** – the unique ID added to the specific user login
- **UserID** – the name of the user
- **lastloginDateTime** – the date and time of last login in the format YYYY-MM-DD HH:mm

Each user is able to set up a wish list:

- **LoginID** – the unique ID added to the login
- **Category of Item** - a value representing one of three categories: Computer, Photography or Security
- **Item code** - the unique Code for each item available
- **Item description** – a brief description of each item available

You have been given a file called **logs.txt** that contains the information of an unknown number of logins and items added to wish lists. Following is the first 11 lines of data from the text file:

```
27#Gerard Hallow#2018-05-20 10:55
27#Gerard Hallow#2018-05-20 10:59#3#CAM02#Outdoor Camera
27#Gerard Hallow#2018-05-20 11:02#1#HDD15#External Hard Drive
28#Simone Selt#2018-05-20 11:08
28#Simone Selt#2018-05-20 11:18#LEN27#Wide angle lens
30#Bruce Hepburn#2018-05-20 12:22
31#Hilary Tinline#2018-05-20 14:40
31#Hilary Tinline#2018-05-20 14:40#1#PRI91#HP LaserJet601
32#Nomsa Sitale#2018-05-20 14:48
32#Nomsa Sitale#2018-05-20 14:48#3#CAB1#Camera Cable
27#Gerard Hallow#2018-05-20 14:68#3#CAM07#Indoor Camera
```

Each line of this file contains the information of a single login or post to wish list.

QUESTION 2

15 MARKS

Use the class diagram below to create a new class called **Login**. This class will be used to create objects that will store the details of one login. The diagram below indicates the properties and methods that are required.

Login
Properties: - loginID : integer - user : string - loginPostDateTime : string
Methods: + Constructor (IID : integer, User : String, loginPostDateTime : string) + getLoginID : integer + getLoginDate : string; + getLoginTime : string; + toString : string

- 2.1 Write code to create a new class called **Login**. (1)
- 2.2 Write code to add the three fields to the class. Ensure that these fields cannot be accessed outside of this class. (3)
- 2.3 Write code to create a **constructor** method that accepts the details of the **Login** class as parameters. These parameters should be used to set the values of the three fields of the **Login** class. (3)
- 2.4 Write code to create an accessor method of the **loginID** field of the Login class. This method should be named appropriately and return the correct type and value for **loginID** property. (1)
- 2.5 Write code to create an accessor method **getLoginDate** to isolate and return the *date* of the **loginPostDateTime** field. (3)
- 2.6 Write code to create an accessor method **getLoginTime** to isolate and return the *time* of the **loginPostDateTime** field. (2)
- 2.7 Write code to create a **toString** method that will return a string representing the post's information. The format is as follows:

```
User <space> "logged in on the" <space> LoginDate <space> "at"
<space> LoginTime
```

Example:

Gerard Hallow logged in on the 2018-05-20 at 10:55 (2)

QUESTION 3

17 MARKS

Use the class diagram below to create a new class called **WishList**. This class will be used to create objects that will store the details of the items added to the user's wish list. The diagram below indicates the properties and methods that are required.

loginID#user#loginPostDateTime#itemCategory#itemID#itemDescription

WishList
Properties: - loginID : integer - user : string - loginPostDateTime : string - itemCategory : integer - itemID : string - itemDescription : string + CATCOMPUTERS : integer = 1 + CATPHOTOGRAPHY : integer = 2 + CATSECURITY : integer = 3
Methods: + Constructor (iD : integer, user : String, loginPostDateTime : String, itemCat : integer, itemId : string, itemDescription : string) + getCategoryType : String + getLoginID : string; + toString : string

- 3.1 Write code to create a new class called **WishList**. (1)
- 3.2 Write code to add the six fields to the class. Ensure that these fields cannot be accessed outside of this class. (3)
- 3.3 Write code to create three class constants as integers that will represent each of the three categories of items sold by the online store. The values for these categories are given in the class diagram. (2)
- 3.4 Write code to create a **constructor** method that accepts the details of the **WishList** class as parameters. These parameters should be used to set the values of the six fields of the WishList class. The **itemCategory** type is an integer. (3)
- 3.5 Write code to create a method called **getCategoryType** which returns a string. This method should test the integer value of the CategoryType field and return a Category description as a string.
 - For CategoryType with a value of CATCOMPUTERS, return "Category: Computers" as a string
 - For CategoryType with a value of CATPHOTOGRAPHY, return "Category: Photography" as a string
 - For CategoryType with a value of CATSECURITY, return "Category: Security" as a string
 (3)

- 3.6 Write code to create an accessor method to create a **getLoginID** property of the **WishList** class. This method should return the correct type and value for the **loginID** property. (1)
- 3.7 Write code that will create a **toString** method that will return a string. The string should be returned in the following format:

```
CategoryType:<newline>
```

```
User <space> "added" <space> itemDescription <space> "Item Code:"  
<space> itemID <space> "at" <space> loginPostDateTime
```

Example:

```
Security:
```

```
Gerard Hallow added Outdoor Camera Item Code: CAM02 at 2018-05-20
```

```
10:59 (4)
```

QUESTION 4

22 MARKS

- 4.1 Write code to create a new class called **WishListManager**. (1)
- 4.2 Write code to declare the following TWO arrays and TWO counter variables.
- An array to store up to 50 **Login** objects and a counter to keep track of the number of Login objects.
 - An array to store up to 200 **WishList** objects and a counter to keep track of the number of WishList objects. (4)
- 4.3 Write code to create a constructor method that will read in the contents of the **logs.txt** file and perform the following:
- Each line of the text file contains one login or one addition to a wishlist.
 - Check if the file exists and display an appropriate message if it does not exist.
 - Read each line of the file.
 - Each line will either be a login or a post to the wishlist.
 - Create the appropriate **Login** or **WishList** object depending on the number of data items in the line of text.
 - Add the **Login** or **WishList** object to the appropriate array. (12)
- 4.4 Write code called **toString**. This method should return a list of all the **Login** objects followed by a list of all **WishList** objects each on a separate line as a String. Use the object's **toString** methods created in question 2 and question 3. (5)

QUESTION 5

3 MARKS

- 5.1 Create a user interface called **ShopUI** that will allow simple output. (1)
- 5.2 Declare and instantiate a **WishListManager** object at the appropriate place in the code. (1)
- 5.3 Write code that will display all the Login and Wishlist objects by calling the appropriate method in the **WishListManager** code. (1)

QUESTION 6

12 MARKS

You are required to write code to create a new class called **LoginWishLists**. This class will store the details of a **Login** object combined with an array of associated **WishList** objects.

- 6.1 Declare a new class called **LoginWishLists** (1)
- 6.2 The **LoginWishLists** class must store details of each client that has logged in, as well as their additions to their wish lists. Declare attributes that will store this data. Each **LoginWishLists** object will only store the details of a single login object and an array of **WishList** objects associated with the single login. You may reuse existing classes instead of redeclaring attributes. (3)
- 6.3 Write code to create a constructor method for this class that will initialise both attributes of this class. (4)
- 6.4 Write a **toString** Method that will return a string with the information of the **Login** and its associated **WishList** items.

```
<Login details><newline>
"Wish List:"<newline>
<tab><tab><tab><WishListDetails><newline>
<tab><tab><tab><WishListDetails><newline>
<tab><tab><tab><WishListDetails><newline>
```

Example:

```
Gerard Hallow logged in on the 2018-05-20 at 10:55
Wishlist:
Security:
Gerard Hallow added Outdoor Camera Item Code: CAM02 on the 2018-
05-20 at 10:59
Computers:
Gerard Hallow added External Hard Drive Item Code: HDD15 on the
2018-05-20 at 11:02
Security:
Gerard Hallow added Indoor Camera Item Code: CAM07 on the 2018-05-
20 at 14:68 (4)
```

QUESTION 7

11 MARKS

Add code to the **WishListManager** class that will populate an array of **LoginWishLists** objects and display the contents of the array.

- 7.1 Add a method called **combineLoginWishLists** to the **WishListManager** class that will populate an array of **LoginWishLists** objects. Each object in the array should store the details of one **Login** object in the original array and an array of associated **WishList** objects.
- Use the **getLoginID** method of the **WishList** class to check whether a **WishList** is related to a **Login**.
 - The method should use relevant **toString** methods to return a string as shown in 7.2 (10)
- 7.2 Display the results in your **OnLineShopUI** method. (1)

A section of the correct output is as follows:

```
Gerard Hallow logged in on the 2018-05-20 at 10:55
Wishlist:
Security:
Gerard Hallow added Outdoor Camera Item Code: CAM02 on the 2018-
05-20 at 10:59
Computers:
Gerard Hallow added External Hard Drive Item Code: HDD15 on the
2018-05-20 at 11:02
Security:
Gerard Hallow added Indoor Camera Item Code: CAM07 on the 2018-
05-20 at 14:68
Simone Selt logged in on the 2018-05-20 at 11:08
Wishlist:
Photography:
Simone Selt added Wide Angle Lens Item Code: LEN27 on the 2018-
05-20 at 11:18
```