# _LITfinalLOGO

# SUMMER EXAMINATIONS 2014

**Tuesday, 13th May 2014, 9.30 a.m. – 11.30 a .m.**

**KSDEM\_8\_Y1**

**Course:** Bachelor of Science (Hons) in Software Development

**Year:** One

**Subject:**  Programming

**Time Allowed:** 2Hours

**Instructions: 1.** You **MUST** answer **Q1**

Answer **ANY OTHER TWO** questions.

**2.** Marks for **Q.1** are **40 marks**.

All other questions are **30 marks.**

**3.** Start each question on a new page.

**4.** Write the question number at the top of each page.

**5.** Circle the numbers of the questions you answer at the front of your answer book.

**Additional Attachments Exam Materials to accompany this paper:**

### A. Attachment A for Q.5

**Internal Examiners: External Examiners:**

Tom Costello Mr Brian Gillespie

**Q. 1**

1. What would be the value of z displayed by each of the following code fragments
   1. **int x = 4, y = 12, z;**

**z = x + y / x;**

**cout << z << endl;**

**(4 marks)**

1. **double x = 5;**

**int y = 4, z;**

**z = x / y;**

**cout << z << endl;**

**(4 marks)**

1. **int x = 6, y = 8, z;**

**x \* = 3;**

**z = x % y;**

**cout << z << endl; (4 marks)**

1. Re-write each of the following code fragments correcting all syntax errors
   1. **first + second = total;**

**if first == second**

**cout << Values are the same\n;**

**cout >> Total is double of either value;**

**else**

**cout << Values are not the same\n;**

**cout << Total is less than the double of the larger value\n;**

**(4 marks)**

**Q.1 contd.**

1. **int option;**

**cin << option;**

**switch option**

**{**

**while 1**

**cout << “Option 1\n”;**

**break;**

**while 2**

**cout << “Option 2\n”;**

**break;**

**else**

**cout << “Invalid option\n”;**

**}**

**(4 marks)**

1. **// function definition**

**void MultTable (int value);**

**{**

**int index = 1**

**while (index <= value)**

**{**

**result = index \* value;**

**cout << index << “ x ” << value << “ = ” << endl;**

**index**

**}**

**}**

**(4 marks)**

**Q.1 contd.**

1. What values will be displayed when each of the following code fragments is executed
   1. **for (int x = 45; x > 0; x - = 15 )**

**{**

**cout << x << endl;**

**}**

**(3 marks)**

* 1. **for ( int row = 0; row < 5; row++ )**

**{**

**for ( int col = 0; col <= 2; col++)**

**cout << row + col << “ “;**

**cout << endl;**

**}**

**(5 marks)**

* 1. **int x = 48;**

**while ( x > 0 )**

**{**

**cout << x << endl;**

**x /= 4;**

**}**

**(3 marks)**

* 1. **for ( int x = 1; x < 30; x + = 3 )**

**{**

**cout << x << “ “;**

**if ( x % 13 = = 0 )**

**break;**

**}**

**(5 marks)**

**(Total 40 Marks)**

**Q. 2** A student’s marks in an exam are converted to a grade according to the table below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mark** | **100 - 85** | **84-70** | **69-60** | **59-50** | **49-40** | **39-0** |
| **Grade** | **A** | **B** | **C** | **D** | **E** | **F** |

A file called “Marks.dat” contains a list of K-numbers and marks for a group of 20 students, one student per row.

Write a program that will

1. read the contents of the file “Marks.dat” into two 1-dimensional arrays, one for the K-numbers and one for the marks. **(10 marks)**
2. populate a third 1-dimensional array with the matching grades based on the table above **(14 marks)**
3. write the K-numbers and grades out to a file called “Grades.dat”.

**(6 marks)**

**(Total 30 Marks)**

**Q. 3** The data relating to a black and white bitmapped image may be stored as a

two dimensional array of integer values, where each value represents the colour of the corresponding pixel. Two such sets of data for two 50 pixel by 20 pixel images are stored in two files called “Picture1.dat” and “Picture2.dat”.

Two images are regarded as identical if at least 95% of corresponding pixel values are the same .

Write a program that will

1. read the contents of the two files into two 2-D arrays. **(12 marks)**
2. then compare the contents of the arrays and count the number of elements that differ **(10 marks)**
3. report whether or not the two images are identical. **(8 marks)**

**(Total 30 Marks)**

**Q. 4** A **class CItem** is added to a project for an application called **ShoppingApp** that is to be used to manage the sale of items.

An item has a four digit code, a description, a standard price and a percentage discount rate. Preferred customers may purchase an item at a discounted price based on the discount rate.

The code below shows a definition of the class.

**class CItem**

**{**

**private:**

**int m\_Code;**

**string m\_Description;**

**double m\_StdPrice;**

**double m\_DiscountRate;**

**public:**

**CItem(void);**

**CItem(int code, string description);**

**CItem(int code, string description, double price, double discountRate);**

**void SetStdPrice(double price);**

**void SetDiscountRate(double discountRate);**

**int GetCode(void);**

**string GetDescription(void);**

**double GetStdPrice(void);**

**double GetDiscountRate(void);**

**bool HasCode(int searchCode);**

**void Display(void);**

**double CalculateDiscount(void);**

**};**

Write definitions for all the member functions of the class

**a)** The first constructor

**CItem(void)**

should initialize the description data member to an empty string and

the other data members to zero

**(4 marks)**

1. The second constructor

**CItem(int code, string description)**

should initialize the appropriate data members to the values passed as arguments and the other data members to zero

**(4 marks)**

1. The third constructor

**CItem(int code, string description, double price, double discountRate)**

should initialize the appropriate data members to the values passed as arguments

**(4 marks)**

1. The **Set** functions should set the appropriate data members

**(2 x 1 marks)**

1. The **Get** functions should return the appropriate data members

**(4 x 1 marks)**

1. The function **HasCode** checksfor a matching code in the object

**(3 marks)**

1. The function **Display** should display the code , description, discount rate and the standard price of an item

**(3 marks)**

1. The function **CalculateDiscount** will calculate and return thediscount available based on the standard price and the discount rate.

**(6 marks)**

**(Total 30 Marks)**

**Q. 5** The application **ShoppingApp** mentioned in **Q.4** is designed to deal with a price list consisting of a maximum of 100 items as indicted by the following global code, where **numItemsInList** contains the actual number of items:

**CItem priceList[100];**

**int numItemsInList;**

The global data will be made persistent by reading from and writing to a file each time the application is opened and closed.

Each time a new item is added to the system the **numItemsInList** variable will be incremented.

The application will display a menu driven interface that will allow the user to access this global collection. Menu options and associated callback functions will allow a user to :

1. add a new item to the list by entering the item code and description of the item. This should cause a **class CItem** object to be instantiated and added to the collection.

The callback function has a prototype **void DoAddItem(void).**

Write a definition for this function.

**(8 marks)**

1. select an item by entering the item code and then set the standard price for that item.

The callback function has a prototype **void DoEnterItemPrice(void)**

Write a definition for this function.

**(10 marks)**

1. calculate and display the total cost for a purchase if the user enters the item code, numbers of items and preferred customer status ( Y or N ).

The callback function has a prototype **void DoPurchaseCost(void)**

Write a definition for this function.

**(12 marks)**

**(Total 30 Marks)**