# _LITfinalLOGO

# SUMMER EXAMINATIONS 2013

**KSDEM\_8\_Y1**

**Tuesday, 14th May 2013, 14.30 p.m. – 16.30 p.m.**

**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 = 3, y = 9, z;**

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

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

**(4 marks)**

1. **int x = 5, y = 13;**

**double z;**

**z = y / x;**

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

**(4 marks)**

1. **int x = 10, y = 6, z;**

**x - = 2;**

**z = x % y;**

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

1. Re-write each of the following code fragments correcting all syntax errors
   1. **int positiveValue;**

**if { positiveVale >= 0 AND < 10 }**

**cout >> “Value is single-digit\n”;**

**else**

**cout >> “Value is multi-digit\n”;**

**(4 marks)**

**Q.1 contd.**

1. **int option;**

**case ( option)**

**switch 1**

**cout << “Option 1\n”;**

**break;**

**switch 2**

**cout << “Option 2\n”;**

**break;**

**else**

**cout << “Invalid option\n”;**

**(4 marks)**

1. **//function definition**

**void GetResult ( int x, y );**

**{**

**int 0 =result;**

**for ( int i = 1; i <= x; i++)**

**{**

**result += i \* y;**

**}**

**return result;**

**}**

**(4 marks)**

**Q.1 contd.**

1. What values will be displayed when each of the following code fragments is executed
   1. **int factorial = 1;**

**for ( int n = 2; n <= 6; n++ )**

**{**

**factorial \*= n;**

**cout << factorial << endl;**

**}**

**(4 marks)**

* 1. **for ( int row = 0; row < 3; row++ )**

**for ( int col = 0; col < 3; col++)**

**cout << row\*col << endl;**

**(4 marks)**

* 1. **int x = 27;**

**while ( x >= 0 )**

**{**

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

**x /= 3;**

**}**

**(4 marks)**

* 1. **for ( int x = 24; x > 0; x - = 3 )**

**{**

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

**if ( x % 5 = = 0 )**

**break;**

**}**

**(4 marks)**

**(Total 40 Marks)**

**Q. 2** A individual’s body mass index (**BMI**), based on his height (**H**) in metres and his weight (**W**) in kilogrammes, may be calculated using the following formula:

**BMI = W / H2**

The person’s BMI Category may be ascertained from the following table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Category** | **Underweight** | **Healthy** | **Overweight** | **Obese1** | **Obese2** | **Obese3** |
| **BMI** | **<18.5** | **18.5 to <25** | **25 to < 30** | **30 to < 35** | **35 to < 40** | **>= 40** |

Write a program that will

1. allow the user to enter the names, weights and heights for a number of people

**(10 marks)**

1. after the data for an individual is entered, the program should
   1. calculate that individual’s BMI

**(6 marks)**

* 1. should then ascertain his BMI Category

**(8 marks)**

* 1. should then write his name, BMI and BMI Category to a file called “Bmidata.dat”

**(6 marks)**

**(Total 30 Marks)**

**Q. 3** In a test, a participant is asked to select a level (1 to 4) and is then assigned

an appropriate task. On completion of the task, the participant is awarded a

number of points . The participant’s overall score is calculated by multiplying

his level by the points awarded to him for the task.

The data for 6 participants is stored in a file called “Test.dat”, each line

showing a participant’s integer id, the level chosen and the points awarded for

the task.

For example, the file might contain the following data where column 1 is a

player’s id, column 2 is the level selected by that player and column 3 is the

points attained for the task

1 2 8

2 3 5

3 3 6

4 4 4

5 2 9

6 4 2

Write a program that will

1. read the file and store its’ contents in a 2-D array

**(10 marks)**

1. scan the array to calculate the highest score attained and then display the highest score.

**(11 marks)**

1. scan the array and display the id(s) of the participant(s) that attained the highest score

**(9 marks)**

**(Total 30 Marks)**

**Q. 4** A board game involves a representation of a pair of named players moving on

a 9 by 9 grid. A player’s position is specified by a row and column pair of

values. The bottom left position is specified as 1, 1 and the top right by 9, 9.

A player enters the game at position 5, 5.

During the game the player may only move horizontally or vertically and may

not move diagonally. In a move the player must proceed in a specified

direction which may be one of four directions: up, down, left or right.

In a single move, the player moves a specified number of grid positions, up to

a maximum of 6 positions. If in the move the player reaches a board edge he

will rebound for the remainder of the move. A players move is specified by a pair of values, a character which is either ‘U’, ‘D’ ‘R’ or ‘L’ and an integer in the range 1 to 6.

In the game, the player gains points accumulated by the sum of his row and

column values at the end of each move.

The game ends when one of the players in a round has accumulated 100 or

more points.

In a project that is to implement the game, the two classes defined below in

the files “Position.h” and “Player.h” are used.

Write an implementation file for each of the classes.

**//File: Position.h**

**class CPosition**

**{**

**private:**

**int m\_Row;**

**int m\_Col;**

**public:**

**CPosition(int row, int col);**

**int GetRow(void);**

**int GetCol(void);**

**void ChangeRow(int amount);**

**void ChangeCol(int amount);**

**};**

1. The constructor function should initialize the data members

**(4 marks)**

1. The Get functions should return the appropriate data member

**(2 x 1 mark)**

1. The member functions **ChangeRow** and **ChangeCol** should change the state of the data members by the **amount**, implementing rebound as appropriate

**(4 marks x 2)**

**//File: Player.h**

**#include "Position.h"**

**using namespace std;**

**class CPlayer**

**{**

**private:**

**string m\_Name;**

**CPosition m\_Position;**

**int m\_Score;**

**public:**

**CPlayer(string name, CPosition position);**

**void Show(void);**

**void Move(char direction, int amount);**

**int GetScore(void);**

**};**

1. The constructor function should initialize the data members

**(4 marks)**

1. The member function **Move** should change the state of the **m\_Position** and the **m\_Score** data members as appropriate.

**(8 marks)**

1. The member function **Show** should display the current values of **m\_Row** and **m\_Col**

**(2 marks)**

1. The member function **GetScore** should return the current value of **m\_Score**

**(2 marks)**

**(Total 30 Marks)**

**Q. 5** Attachment A contains a skeleton of the application (with comments and missing code) that will implement the game specified in **Q. 4**.

The code includes the definition of a function **GenerateMove** that will randomly assign values to a character direction and an integer amount

Re-write the code for the function

**int \_tmain(int argc, \_TCHAR\* argv[ ])**

Insert appropriate additional code where the comment *MISSING CODE x* is in the function as indicated below

1. *MISSING CODE A* should instantiate a **CPosition** object, initialized as appropriate

**(2 marks)**

1. *MISSING CODE B* should allow the user to enter the names of the players

**(2 marks)**

1. *MISSING CODE C* should instantiate two **CPlayer** objects, initialized as appropriate.

**(4 marks)**

1. *MISSING CODE D* should display the position of each player

**(2 marks)**

1. *MISSING CODE E* 
   1. should set up the game rounds loop that should loop until one of the players has attained a score of 100 or more in a round

**(6 marks)**

* 1. Within the loop a random move should be generated for each player, the move should be applied to the player and then the player’s position should be displayed

**(8 marks)**

1. *MISSING CODE E* should display the name of the player who has the highest score

**(6 marks)**

**(Total 30 Marks)**

**Attachment A**

**// boardgame2.cpp : Defines the entry point for the console application.**

**#include "stdafx.h"**

**#include "Position.h"**

**#include "Player.h"**

**using namespace std;**

**//function prototype**

**void GenerateMove(char& direction, int& size);**

**int \_tmain(int argc, \_TCHAR\* argv[])**

**{**

*//initialize the randon number generator*

**srand( time(0) );**

*//set up a starting board position for the players*

*//MISSING CODE A*

*//Get players names*

**string name1,name2;**

*//MISSING CODE B*

*//Instantiate two players*

*//MISSING CODE C*

*//Set up header for display table and show each players starting position*

**cout << "\n\n" << name1 << setw(20-name1.length()) << name2 << endl;**

*//MISSING CODE D*

*//play rounds of game until one player reaches a score of 100*

*//MISSING CODE E*

*//display the name of the player who has won the game*

*//MISSING CODE F*

**return 0;**

**}**

*//Function will generate a random direction and amount for a players move*

**void GenerateMove(char& direction, int& amount)**

**{**

**int d = rand() % 4 + 1;**

**switch ( d )**

**{**

**case 1:**

**direction = 'U';**

**break;**

**case 2:**

**direction = 'D';**

**break;**

**case 3:**

**direction = 'R';**

**break;**

**case 4:**

**direction = 'L';**

**break;**

**}**

**amount = rand() % 6 + 1;**

**}**