

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION**WINTER SEMESTER, 2017-2018****DURATION: 1 Hour 30 Minutes****FULL MARKS: 75**

CSE 4301: Object Oriented Programming

Programmable calculators are not allowed. Do not write anything on the question paper.

There are 4 (four) questions. Answer any 3 (three) of them.

Figures in the right margin indicate marks.

1. a) Figure 1 presents a class called **MatrixElement**. The class represents an element in a Matrix. The properties **row** and **col** represents the row number and column number of the element in a matrix, respectively. The row and column numbers start from 0. The property **val** represents the integer value of the element. 15
- Create a class called **Matrix** that stores the elements of a matrix privately in the form of **MatrixElement** objects. Create a public method inside **Matrix** class that asks the user to provide the dimensions of the matrix (number of rows and number of columns) and based on the user input dynamically create required **MatrixElement** objects. For each of the **MatrixElement** objects, set the **row**, **col** and **val** properties by taking input from the user. It should be noted that you cannot change the body of **MatrixElement**.

```
class MatrixElement{
    int row, col, val;
public:
    MatrixElement() :row(0), col(0), val(0) {}

    MatrixElement(int row, int col, int val) :row(row),
col(col), val(val) {}

    int getRow() {
        return row;
    }

    int getCol() {
        return col;
    }

    int getVal() {
        return val;
    }

};
```

Figure 1: MatrixElement class

- b) Update the class **Matrix**, created in Question 1(a), to add a public method that takes one **Matrix** object as parameter and multiplies the **MatrixElement** objects from the **Matrix** passed as parameter with its own **MatrixElement** objects. You have to at first check for dimension of the two matrices before multiplying them. To do that, you can update the class **Matrix** to include the number of rows and columns for each matrix. 10

2. Create a class called **StudentInfo** where the details of a student's academic records are stored. In the above mentioned class, the private properties will include the name and ID of the student. Moreover, a student can take any number of courses and the results of the courses are to be kept in an integer array inside **StudentInfo** class. All these properties need to be private and can only be accessed outside of **StudentInfo** class by **CalculateAvgResult** method from **ResultService** class and through the parameterized constructor of the **StudentInfo** class. It should be noted that the properties of the **StudentInfo** class can only be set once through this its constructor as parameters and the values should be immutable. **CalculateAvgResult** method calculates the average result of a given student. Moreover, **ResultService** include another method called **SortStudents** which takes an array of students as parameter and sorts them according to their average results in ascending order. Finally, **SortStudents** prints the names of the students according to the sorted list.

Note: The **interface** and **implementation** for both the classes should be in separate files with each file marked elaborately along with their names. Moreover, you should include a main function in a separate file to demonstrate your implementation. Each file should refer to all required header and library files according to necessity.

3. a) Create a class called **Rectangle**. The class has attributes **length** and **width**, each of which defaults to 1. It has member functions that calculate the **area** and **perimeter** of the rectangle. It has *set* and *get* functions for both **length** and **width**. The *set* function should verify that the length and width of the rectangle are floating point numbers larger than 0.0 and smaller than 20.0. 10
- b) Create a more sophisticated rectangle class than the one in question 3(a), which stores the Cartesian coordinates of the four corners of the rectangle. The constructor calls a set function that accepts four sets of coordinates. Verify that each of these coordinates falls into the first quadrant and none of the x or y coordinates is larger than 20.0. Also verify that the supplied coordinates in fact creates a rectangle. If the length and width are same for the rectangle, classify the rectangle as a Square. 10 + 5
- Hint:** Use Pythagorean Theorem $a^2 + b^2 = c^2$. Distance between two points is $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.
4. a) How Structured programming is different from Object Oriented Programming? 5
- b) Write short notes on 4×2
- Separation of Concerns
 - Encapsulation
 - Polymorphism
 - Inheritance
- c) Consider the problem in Question 2. If **StudentInfo** class included a destructor that would free up memory for the student's name and results array, will there be any problem in executing the program. Explain your answer. 12