ISTANBUL SABAHATTIN ZAIM UNIVERSITY

BIM227 – DATA STRUCTURES

ASSIGNMENT – 01

Assoc. Prof. Dr. Jawad RASHEED

2023-2024
FALL SEMESTER

Nov 05, 2023

Question	1	2			TOTAL
Maximum Score	30	70			100
Student Score					

Instructions

- This assignment is equal to one quiz.
- Uncommented, Cheated or Plagiarized answers will get ZERO.
- All the codes should be error-free, and each line and function/class must be commented on properly.
- Your code for each question should be in a separate folder. The folder should not contain files other than *.cpp* and *.h*
- You must zip (compress) the folders in a single zipped file and submit them any time before November 12, 2023, 23:00 hours through the OYS/Kampus system.
- *Submissions through email will not be entertained.*

Questions

Question 1: Write a C++ program that calculates the square of a number using a function. The program should do the following to calculate the square:

i. Takes the number as an input from the user and stores it in an *int* variable.

ii. Write a function ***calculateSquare*** that has an integer pointer (***int****) as an argument to take the value, calculates its square, and then returns the calculated value through a pointer (in other words, the function return type is also ***int****). For your convenience, the function prototype is written below:

int* calculateSquare (int*);

Question 2: Write another C++ program that does the following:

i. Implement a class called ***myMatrix***. It has the following

a. *private* member variables:

- *int r* // for the number of rows in the matrix
- *int c* // for the number of columns in the matrix
- *int **ptr* // for dynamic declaration of the matrix

b. constructor

myMatrix(int r, int c) // create a matrix using ptr based on values of r and c.

c. destructor

d. *public* member functions:

- *initializeMatrix ()* // store the values entered by the user.
- *getspecificElement(int x, int y)* // returns the value stored at [x,y]th location of the matrix.
- *putElementatSpecificLocation (int x, int y, int value)* // stores the value at [x,y]th location of the matrix. The function should return true if stored successfully or false if not.

- *calculateRowSum ()* // calculates the average value of each row and only returns the highest average value.
- *printMatrix ()* // prints all the values stored in the matrix in a fashion shown at the end of this document.

ii. In the ***main*** class function, take values of *row* and *column* from the user and then pass these values by calling the constructor of ***myMatrix*** to create its instance.

Hint: For your convenience matrix looks like the following. It has 3 rows and 4 columns.

2 5 9 7
6 9 1 0
4 2 0 3