NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES ISLAMABAD

Object Oriented Programming (CS 1004)

SPRING 2022 ASSIGNMENT # 3

Due Date ≈ Friday, April 29th, 2022 (05:00 pm)

Instructions

Submission: Combine all your work in one .zip file. Use proper naming convention for your submission file. Name the .zip file as **SECTION_ROLLNUM.zip** (**e.g. M_21i0412.zip**). Your zip file should only contain folder of each question as Q1, Q2… etc. Your submission should not contain any further folders or subfolders. It should only contain .cpp or .h files for each question, if additional files are asked they will be mentioned with each question. Submit .zip file on Google Classroom within the deadline. Failure to submit according to the above format would result in **25% marks deduction**. Submissions on the email will not be accepted.

Plagiarism: Plagiarism cases will be dealt with strictly. If found plagiarized, both the involved parties will be awarded zero marks in this assignment, all the remaining assignments, or even an **F grade** in the course. Copying from the internet is the easiest way to get caught!

Deadline: The deadline to submit the assignment is 29th **April 2022 at 05:00 PM**. Late submission with marks deduction will be accepted according to the course policy shared earlier. Correct and timely submission of the assignment is the responsibility of every student; hence no relaxation will be given to anyone.

Bonus: In case you implement any additional feature which you think is worth of bonus, make it prominent so that we can see it at runtime.

Note:

- *Each question will be graded on the basis of your effort, additional marks will be awarded for using good programming practices, including: memory efficient programs, well-written, good design and properly commented.*
- All programs must be generic.
- You can change the argument, return type and also add new data members in the given structures.
- Follow the given instructions to the letter, failing to do so will result in a zero.

Q1) Number Array Class: Design a class that has an array of floating-point numbers. The constructor should accept an integer argument and dynamically allocate the array to hold that manynumbers. The destructor should free the memory held by the array. In addition, there should be member functions to perform the following operations:

- Store a number in any element of the array
- Retrieve a number from any element of the array
- Return the highest value stored in the array
- Return the lowest value stored in the array
- Return the average of all the numbers stored in the array

Q2) Implementation of Array Class: Your goal is to implement a generic "Array" class. Please also write down the test code to drive your class implementation.

```
class Array{
// think about the private data members...
public:
// provide definitions of following functions...
```

- Array();// a default constructor
- Array(int size);// a parametrized constructor initializing an Array of predefined size
- Array(int *arr, int size);// initializes the Array with an existing Array
- Array(const Array &);// copy constructor
- int getAt(int i);// returns the integer at index [i]
- void setAt(int i, int val);// set the value at index [i]
- Array subArr(int pos, int siz);// returns a sub-Array of size siz starting from location'pos'
- Array subArr(int pos);// returns a sub-Array from the given position to the end.
- int * subArrPointer(int pos, int siz);// returns an array of size siz starting from location'pos'
- int * subArrPointer(int pos);// returns an array from the given position to the end.
- void push_back(int a);// adds an element to the end of the array
- int pop_back();// removes and returns the last element of the array
- int insert(int idx, int val);// inserts the value val at idx. Returns 1 for a successful insertion and -1 if idx does not exists or is invalid. Shift the elements after idx to theright.
- int erase(int idx, int val);// erases the value val at idx. Returns 1 for a successful deletionand -1 if idx does not exists or is invalid. Shift the elements after idx to the left.
- void size();
- int length();// returns the size of the Array
- void clear();//clears the contents of the Array
- int value(int idx);//returns the value at idx
- void assign(int idx, int val);//assigns the value val to the element at index idx
- void copy(const Array& Arr);// Copy the passed Array
- void copy(const int * arr, int siz);// copy the passed array
- void display();// displays the Array
- bool isEmpty();// returns true if the Array is empty
- Array find(int);// returns an Array containing all the indexes of integer being searched
- bool equal(Array);// should return true if both Arrays are same
- int sort();// sorts the Array. Returns true if the array is already sorted

- void reverse()://reverses the contents of the array
- ~Array();// destructor...

}

Q3) Rational Class: Create a class called Rational for performing arithmetic with fractions. Writea program to test your class.
Use integer variables to represent the private data of the class

- Numerator
- Denominator.

Provide a constructor that enables an object of this class to be initialized when it's declared. The constructor should contain default values in case no initializers are provided and should store the fraction in reduced form. For example, the fraction 2/4 would be stored in the object as 2 in the numerator and 4 in the denominator. Provide public member functions that perform each of the following tasks:

- Adding two Rational numbers. The result should be stored in reduced form.
- Subtracting two Rational numbers. The result should be stored in reduced form.
- Multiplying two Rational numbers. The result should be stored in reduced form.
- Dividing two Rational numbers. The result should be stored in reduced form.
- Printing Rational numbers in the form a/b, where a is the numerator and b is the denominator.
- Printing Rational numbers in floating-point format.

Q4) HugeInteger Class: Create a class HugeInteger that uses a 40-element array of digits to storeintegers as large as 40 digits each. Provide member functions: Input, Output, Add and Subtract.
For comparing HugeInteger objects, provide functions:

- isEqualTo
- isNotEqualTo
- isGreaterThan
- isLessThan
- isGreaterThanOrEqualTo
- isLessThanOrEqualTo
- isZero

Each of these is a "predicate" function that simply returns true if the relationship holds betweenthe two HugeIntegers and returns false if the relationship does not hold.

Q5) Trivia Game – In this question you will create a simple trivia game for two players. The program will work like this:

- Starting with player 1, each player gets a turn at answering five trivia questions. (There are a total of 10 questions.) When a question is displayed, four possible answers are also displayed. Only one of the answers is correct, and if the player selects the correct answer he or she earns a point.
- After answers have been selected for all of the questions, the program displays the number of points earned by each player and declares the player with the highest number of points the winner.

In this program you will design a Question class to hold the data for a trivia question. . You will need to write three files (Question.h, Question.cpp and Q2.cpp). The Question class should have member variables for the following data:
A trivia question

- Possible answer #1
- Possible answer #2
- Possible answer #3
- Possible answer #4
- The number of the correct answer (1, 2, 3, or 4)

The Question class should have appropriate constructor(s), accessor, and mutator functions. The program should create an array of 10 Question objects, one for each trivia question. Make up your own trivia questions on the subject or subjects of your choice for the objects.