

NATIONAL UNIVERSITY OF COMPUTER & EMERGING
SCIENCES ISLAMABAD

Object Oriented Programming (CS 1004)
SPRING 2022 ASSIGNMENT # 2

Due Date ≈ Wednesday, April 13th, 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_ROLL-NUM_01.zip** (e.g. **P_21i0412_01.zip**). Your zip file should not contain any 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 13th 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.

Q 1 – You have to create a Class named *Student*. For each student we want to store his/her Roll Number, Name, Batch, number of registered courses, course codes, course names, course grades, CGPA, Degree, and Date of Birth. You will need to write three files (Student.h, Student.cpp and Q1.cpp)

Your implemented class must fully provide the definitions of following class (interface) functions. Please also write down the test code to drive your class implementation. Please note that we will be running your code against our test code and any segmentation faults or incorrect result will result in loss of marks.

```
class Student{
private:
    // think about the private data members...
public:
    // provide definitions of following functions...
    Student();// default constructor
    Student(char *str);
    Student(double);
    Student(const Student &);

    //implement mutators for all private data members
    //implement accessors for all private data members

    //you have to implement the following functions
    // think about the parameters required and return type of the following
    functions
    DisplayCourses();//displays the courses Student is registered in
    DisplayGrade();//display grades in each course
    Display();//displays the transcript
    comparesStudent();//static function that returns the student with better score
    in a given course
    sortStudents();//static function that sorts the students in ascending order on
    the basis of GPA

    ~Student();
};

int main(){
    //write test code
}
```

Q 2: 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.

Q 3: Rational Class – Create a class called Rational for performing arithmetic with fractions. Write a 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. You will need to write three files (Rational.h, Rational.cpp and Q3.cpp)

```
class Rational {
private:
    // private data members...
public:
    // provide definitions of following functions...
    //constructor

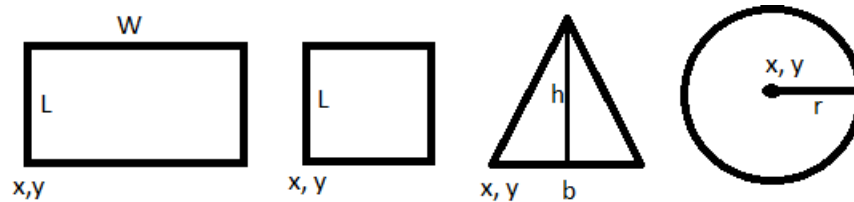
    //implement mutators for all private data members
    //implement accessors for all private data members

    //you have to implement the following functions
    // think about the parameters required and return type of the following
    functions
    Add();//add two Rational numbers. The result should be stored in reduced form
    Subtract();//subtracts two Rational numbers. The result should be stored in
    reduced form
    Multiply();//multiplies two Rational numbers. The result should be stored in
    reduced form
    Divide();//divides two Rational numbers. The result should be stored in reduced
    form
    PrintRationalFormat();//Printing Rational numbers in the form a/b, where a is
    the numerator and b is the denominator
    PrintFloatFormat();//Prints Rational numbers in floating-point format
};

int main(){
    //write your test code here...
}
```

Q 4: Shape Class – Geometrical shapes are classified on the basis of their dimensions. Circle, Triangle, Square and Rectangle are most 2D shapes. You have to write a class Shape which can store these four basic 2D shapes. Remember

these shapes are normally associated with the number of sides they have. You will need to write three files (Shape.h, Shape.cpp and Q4.cpp)



Your implemented class must fully provide the definitions of following class (interface) functions. Please also write down the test code to drive your class implementation. Please note that we will be running your code against our test code and any segmentation faults or incorrect result will result in loss of marks.

```
class Shape{
private:
    // think about the private data members...
public:
    // provide definitions of following functions...
    Shape(); // default constructor

    Shape(int...); // parameterized constructor that will allow to initialize different
    types of instances

    // implement mutators for all private data members
    // implement accessors for all private data members

    // you have to implement the following functions
    // think about the parameters required and return type of the following
    functions
    getArea(); // prints area
    overlapping(); // determines if two Shapes are overlapping or not
    returnLargestShape(); // a static function that returns the largest Shape from the
    array of Shapes, this function should only work for the same shapes
    overlappingShapes(); // a static function that returns an array of Shapes
    overlapping the largest Shape, this function should only work for the same
    shapes
    ~Shape();
};

int main(){
    // you have to create different objects and demonstrate all the functions of
    shape class
    // you to create N Shape objects of same type and give user options to initialize
    them
}
```

Q 5: Implementation of Character Class – Your goal is to implement “Character” class. You will need to write three files (Character.h, Character.cpp and Q5.cpp). Your implemented class must fully provide the definitions of following

class (interface) functions. Please also write down the test code to drive your class implementation. Please note that we will be running your code against our test code and any segmentation faults or incorrect result will result in loss of marks.

```
class Character{
    // think about the private data members...
public:
    //include all the necessary checks before performing the operations in the
    functions
    Character();// a default constructor
    Character(int);// a parametrized constructor
    Character(String); // a parametrized constructor
    void set(char);//set value
    isEqualTo(Character); //Compares two Character objects numerically
    isNotEqualTo(Character);
    isGreaterThan(Character);
    isLessThan(Character);
    isGreaterThanOrEqualTo(Character);
    isLessThanOrEqualTo(Character);
    isAlpha();//checks if the Character is an alphabet
    isNum();//checks if the Character is a number
    isLower();//check for lower case
    isUpper();//check for upper case
    toUpper();//convert a lowercase alphabet to uppercase
    toLower();//convert a uppercase alphabet to lowercase
    String stringValue(); //Returns the value of this character as a string
    Character plus(const Character&); //adds two Characters and return the result
    Character minus(const Character&); // subtracts two Characters and return the
    result
};
```

Q 6: Resource Monitoring – Human resource monitoring is normally done centrally. HR office keeps track of every employee storing basic information along with their leave record, salary details. In this question you will have to write two separate classes: 1) HRManager – that keeps tracks of all employees; 2) Employee – that keeps track of every employee. You will need to write three files (HR.h, HR.cpp and Q6.cpp)

Consider following classes/structures, you can always add additional data member if you feel it will result in better/efficient design:

```
class HRManager { /*Special class with only one instance during execution of the
    program, there is only one HR manager*/

private:
    // think about the private data members...

    // constructors will be private
public:

    // provide definitions of following functions...
```

```
getHR_Instance(); //always returns a pointer/instance to the one and only
instance of HRManager, when called the first time create a record of N employees
Employee* getEmployee(); //returns Employee instance with the given ID
employeeStatus(); //prints the status of all the employees
printLeaveBalance(); //prints the leave balance of all the employees
printSalaryReport(); //prints the salary of all employees
};

class Employee{
private:
    // private data members (id, age, status(idle-0/busy-1), salary, casual leaves
    total, casual leave availed, earned leaves)
    static int empCount; //can be used to assign ID to employees
public:
    // default constructor
    //parameterized constructors (3)
    // copy constructor

    //whenever an employee state is change HRManager instance is called from that
    function

    Employee* setSalary();
    Employee* setStatus();
    Employee* setCasualLeaves();
    AvailCasualLeave(); //Employee avails casual leaves, leaves record will be
    updated and also updated for HRManager using its instance
    AvailEarnedLeave(); //same as above but for earned leaves
    CompareEmployee(Employee); //compares two employees, return the one with higher
    salary
    ~Employee();
};

int main(){
    //you have to create different objects and demonstrate all the functions of
    Employee class
    //you to create N employees dynamically... also using an HRManager instance update
    the HRManager about the number of employees and update their data
}
```
