NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS
CS217 Object Oriented Programming- Fall 2020
ASSIGNMENT- 3
Section (A, B, C, D, and F)
Due Date: Sunday 15th November 2020 at 11:59 pm on Google Classroom

Instructions:

1. Assignments are to be done individually. You must complete this assignment by yourself. You cannot work with anyone else in the class or with someone outside of the class. The code you write must be your own and you must understand each part of coding. You are encouraged to seek help from the instructors through email, on google classroom or individually visiting their respective offices.
2. The AIM of this assignment is to practice with Classes and Structures in C++.
3. No late assignments will be accepted.
4. Displayed output should be well mannered and well presented. Use appropriate comments and indentation in your source code.
5. Plagiarism:

Plagiarism of any kind (copying from others and copying from internet, etc.,) is not allowed. If found plagiarized, you will be awarded zero marks in the assignment. Repeating such an act can lead to strict disciplinary actions and failure in course.

Submission Guidelines:

We will be using auto-grading tools, so failure to submit according to the following format would result in zero marks in the relevant evaluation instrument:

i) For each question in your assignment, make a separate .cpp file e.g. for question 1, make q1.cpp and so on. Each file that you submit must contain your name, student-id, and assignment # on top of the file in the comments.
ii) Combine all your work in one folder. The folder must contain only .cpp files (no binaries, no exe files etc.,).
iii) Run and test your program on a lab machine before submission.
iv) Rename the folder as ROLL-NUM_SECTION (e.g. 19i-0001_B) and compress the folder as a zip file. (e.g. 19i-0001_B.zip).
v) Submit the .zip file on Google Classroom within the deadline.
vi) Submission other than Google Classroom (e.g. email etc.) will not be accepted.
vii) The student is solely responsible to check the final zip files for issues like corrupt file, virus in the file, mistakenly exe sent. If we cannot download the file from Google classroom due to any reason, it will lead to zero marks in the assignment.

Question 1:

Implement a structure Employee. An employee has a name (a char *) and a salary (a double). Write a default constructor, a constructor with two parameters (name and salary), and methods **char* getName()** **double getSalary()** to return the name and salary.
Write a small global function **TestEmployee()** to test your structure.

```
Creating a new employee.
Please type the name:
Larry Bird
Please specify the salary: 200000
New employee has been created.
Name of employee: Larry Bird
Salary: 200000.0
Thank you for testing structure Employee.
```

Question 2:

Implement a structure Car with the following properties. A car has a certain fuel efficiency (measured in miles per gallon or liters per km **pick one**) of type **float** and a certain amount of fuel in the gas tank of type **float**. The efficiency is specified in the constructor, and the initial fuel level is 0. Supply a method **drive (float)** that simulates driving the car for a certain distance, reducing the fuel level in the gas tank, and methods

float **getFuelLevel()** returning the current fuel level, and
void **tank(float)**, to tank up.

Sample usage of the structure:

```
 void main() {
  Car myBeemer(29);
  cout<<myBeemer.getFuelLevel()<<endl;
  myBeemer.tank(20);
  cout<<myBeemer.getFuelLevel()<<endl;
  myBeemer.drive(100);
  cout<< myBeemer.getFuelLevel()<<endl;
}
```

Should produce:

```
0.0
20.0
16.551724137931036
```

Question 3:

Implement a structure Circle (think of its data members) that has methods

- float **getArea()** and
- float **getCircumference()**

In the constructor, supply the radius of the circle.

```
Please specify the radius of your circle: 1.0
Circle created.
Area: 3.141592653589793
Circumference: 6.283185307179586
Good-bye!
```

Question 4:

Define a class FlightInfo in C++ with following description:

Private Members
A data member **FlightNumber** of type integer
A data member **Destination** of type char*
A data member **Distance** of type float
A data member **Fuel** of type float

A member function void **calFuel()** to calculate the value of Fuel as per the following criteria and set its corresponding data member

Distance	Fuel
<=1000	500
more than 1000 and <=2000	1100
more than 2000	2200

Public Members
A function void **feedInfo()** to allow user to enter values for Flight Number, Destination, Distance & call function void **calFuel()** to calculate the quantity of Fuel
A function void **showInfo()** to allow user to view the content of all the data members
A function float **getFuel()** that returns the current fuel value.

Question 5:
Implement a class Employee2. An employee has a name (a char *) , HourlyWage (float) , WorkedHours(float) and ExtraHours(float).

Write a function
float **wageCalculator()**

that reads in the name and hourly wage of an employee. Then ask how many hours the employee worked in the past week. Be sure to accept fractional hours. Compute the pay. Any overtime work (over 40 hours per week) is paid at 150 percent of the regular wage. Print a paycheck for the employee.

```
Please enter employee's name then press Enter : Larry Bird
Please enter hourly wage then press Enter : 12.50
Please enter hours worked then press Enter: 10
 Paycheck for employee Larry Bird

 Hours worked: 10.0
 Hourly wage: 12.5

 Total payment: 125.0


Please enter employee's name then press Enter : Michael Jordan
Please enter hourly wage then press Enter : 10
Please enter hours worked then press Enter: 50
 Paycheck for employee Michael Jordan

 Hours worked: 50.0
 Hourly wage: 10.0

 Overtime hours: 10.0
 Overtime hourly wage: 15.0

 Total payment: 550.0
```

Question 6:

Implement a class Address. An address has

- a **HouseNumber (int)**,
- a **street (int)**,
- an optional **ApartmentNumber (int)**,
- a **city(char*)**,
- a **state(char*)**
- **PostalCode(int)**.

write two constructors:

- one with an ApartmentNumber
- and one without.
- Write a *void* **print()** function that prints the address with the street on one line and the city, state, and postal code on the next line.
- Write a method *bool* **compareTo()** that tests whether one address comes before another when the addresses are compared by postal code

Question 7:
Implement a class Account. An account has data member:

- a **balance(float)**,

and member functions
void **deposit(float)** add money
bool **withdraw(float)** withdraw money after checking conditions
float **inquire ()** returns the current balance.

- Pass a value into a constructor to set an initial balance.
- If no value is passed the initial balance should be set to $0.
- Charge a $5 penalty if an attempt is made to withdraw more money than available in the account.

Question 8

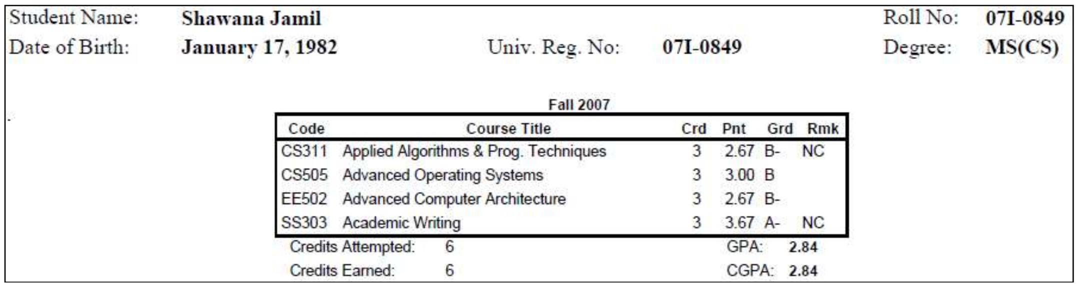
Create a Class named *Student* with following private data members

- Roll Number(char *)
- Name(char *)
- Batch(int)
- An Array named Courses_Code(int) of length 5, containing course code of the registered courses
- An Array named Courses_Name(char *) of length 5, containing course name of the registered courses
- An Array named Courses_Grades(char) of length 5, containing course grades of the registered courses
- CGPA(float)
- Degree(char *)
- Date of Birth(char *)

The class should have following functions

- **setValues(),** to set values of the variables
- A default constructor to initialize all data members to some initial value
- An overloaded constructor which is passed values of all data members as argument
- 9 different functions to update/change value of each of the data member
- A display function to display transcript of the student in following format (including box)

Create an instance of the above structure in function named studentDemo void **studentDemo()** and demonstrate use of all member function of the structure.



Student Name:	**Shawana Jamil**			Roll No:	**07I-0849**
Date of Birth:	**January 17, 1982**	Univ. Reg. No:	**07I-0849**	Degree:	**MS(CS)**

Fall 2007

Code	Course Title	Crd	Pnt	Grd	Rmk
CS311	Applied Algorithms & Prog. Techniques	3	2.67	B-	NC
CS505	Advanced Operating Systems	3	3.00	B	
EE502	Advanced Computer Architecture	3	2.67	B-	
SS303	Academic Writing	3	3.67	A-	NC

Credits Attempted: 6 GPA: **2.84**
Credits Earned: 6 CGPA: **2.84**

Question 9:

Write the definition for a class called **Rectangle** that has floating-point data members length and width. The class has the following member functions:

- **void setLength(float)** to set the length data member
- **void setWidth(float)** to set the width data member
- **float perimeter()** to calculate and return the perimeter of the rectangle
- **float area()** to calculate and return the area of the rectangle
- **void show()** to display the length and width of the rectangle
- **int sameArea(Rectangle)** that has one parameter of type Rectangle. sameArea() returns 1 if the two Rectangles have the same area, and returns 0 if they don't.

Question 10:

Implementation of Array Class Your goal is to implement a generic “**Array**” class. Your implemented class must fully provide the definitions of following class (interface) functions given in the code snippets below:

```cpp
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 the right.
int erase(int idx, int val);// erases the value val at idx. Returns 1 for a successful deletion and -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(const 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...
};
```