NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS
CS-1004 Object Oriented Programming
Assignment No. 2
Section (A, B, C and D)

Submission Information

Submission Deadline: **March 19th, 2023 at 10:00 AM**. You are supposed to submit your assignment on GOOGLE CLASSROOM (CLASSROOM TAB not lab). Only ".ZIP" files are acceptable. Other formats should be directly given ZERO. Correct and timely submission of the assignment is the responsibility of every student, hence no relaxation will be given to anyone. Late Submission policy will be applied as described in course outline.

Tips: For timely completion of the assignment, start as early as possible.

Plagiarism: Plagiarism is not allowed. If found plagiarized, you will be awarded zero marks in the assignment (copying from the internet is the easiest way to get caught).

Instructions:

Dear students, we will be using auto-grading tools, so failure to submit according to the format below 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 ROLL-NUM_SECTION_Q#.cpp (22i-0001_A_Q1.cpp) and so on. Each file that you submit must contain your name, student-id, and assignment # on top of the file in 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. 22i-0001_A) and compress the folder as a zip file. (e.g. 22i-0001_A.zip). Do not submit .rar file.

V.   Submit the .zip file on Google Classroom within the specified 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.

VIII. Displayed output should be well mannered and well presented. Use appropriate comments and indentation in your source code.

IX.   **Total Marks: 100.**

Learning Objective

The AIM of this assignment is to have hands-on classes.

Warning:

- If there is a syntax error in the code, zero marks will be awarded in that part of the assignment.
- Your code must be generic and handle all errors and exceptions.

Question 1:

Implement a class Employee. An employee has a name (a char *) and a salary (a double). Write a default constructor with two parameters (name and salary), and methods **char* getName()**
double getSalary() to return the name and salary.
Implement driver function to implement your class.

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 Class Employee.

Question 2:

Implement a class 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 class:
```
 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:

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 4:
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 5:

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 6:
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 7:

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 class in function named studentDemo void **studentDemo()** and demonstrate use of all member function of the class.

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 8:
Implementation of String Class – Implementation of String Class Your goal is to implement a generic "String" class using cstring, i.e. you will need to write three files (string.h, string.cpp and Q8.cpp). Your implemented class must fully provide the definitions of following class (interface) functions. You 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.

```cpp
class String{
```

```
private:
  // think about the private data members...
public:
  // provide definitions of following functions...
  String();// default constructor
  String(char *str);// initializes the string with constant cstring
  String(const String &);// copy constructor
  String(int x);// initializes a string of pre-defined size
  char getAt(int i);// returns the character at index [x]
  void setAt(int i, char c);// set the character at index [x]
  String substr(int pos, int len);// returns a substring of length len from 'pos'
  String substr(int pos);// returns substring from the given position to the end.
  void append(char a);// append a char at the end of string
  void append(String str );// append a String at the end of string
  void append(char *str );// append a constant c string at the end of string
  int length();// returns the length of string
  char * tocstring();// converts a String to c-string
  void display();// displays the string ..
  bool isEmpty();// returns true if string is empty..
  void copy(const String&);// Copy one string to another ...
  void copy(const char *);// copy cstring to String...
  int find(char);// returns the index of character being searched.
  bool equal(String);// should return true if both strings are same
  int stoi();// function for converting a string to integer.
  ~String();// destructor...
};
```

Question 9:

Implementation of Integer Class – Your goal is to implement "Integer" class. You will need to write three files (Integer.h, Integer.cpp and Q9.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.

```cpp
class Integer{
    // think about the private data members...
    public:
        //include all the necessary checks before performing the operations in the
        functions
        Integer();// a default constructor
        Integer(int);// a parametrized constructor
        Integer(String); // a parametrized constructor
        void set(int);//set value
        int get()const;//get value at (i,j)
        int bitCount(); //Returns the number of one-bits in the 2's complement binary
        int compareTo(Integer); //Compares two Integer objects numerically.
        double doubleValue(); //Returns the value of this Integer as a double.
        float floatValue(); //Returns the value of this Integer as a float.
        Integer plus(const Integer&); //adds two Integers and return the result
        Integer minus(const Integer&); // subtracts two Integers and return the result
        Integer multiple(const Integer&); //multiplies two Integers and return the result
        Integer divide(const Integer&); //divides two Integers and return the result
        static int numberOfLeadingZeros(int i); /*Returns the number of zero bits
        preceding the highest-order ("leftmost") one-bit in the two's complement binary
        representation of the specified int value.*/
        static int numberOfTrailingZeros(int i); /*Returns the number of zero bits
        following the lowest-order ("rightmost") one-bit in the two's complement binary
        representation of the specified int value.*/
        static String toBinaryString(int i); //Returns string representation of i
        static String toHexString(int i); //Returns string representation of i in base16
        static String toOctString(int i); //Returns string representation of i in base 8
};
```

Question 10:

Implementation of Array Class -Your goal is to implement a generic "**Array**" class. You will need to write three files (Array.h, Array.cpp and Q10.cpp). 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...
};
```