NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES ISLAMABAD CAMPUS
OBJECT ORIENTED PROGRAMMING (CS103) - SPRING 2023 ASSIGNMENT-2

Due Date: March 28, 2023(11:59 PM)

Instructions:

- *Make sure that you read and understand each and every instruction.*
- *Create each problem solution in a separate .cpp file, i.e. you must name the file containing solution of* ***Q1 as 'q1.cpp', Q2 as 'q2.cpp' and Q3 as 'q3.cpp'.***
- *Combine all your work in one .zip file.*
- *Name the .zip file as ROLL-NUM SECTION.zip (e.g. 22i-0001 B.zip).*
- *Submit the .zip file on Google Classroom within the deadline.*
- *Start early otherwise you will struggle with the assignment.*
- *You must follow the submission instructions to the letter, as failing to do so will get you a zero in the assignment.*
- *All the submitted evaluation instruments (quizzes, assignments, lab work, exams, and the project) will be checked for plagiarism. If found plagiarized, both the involved parties will be awarded zero marks in the relevant evaluation instrument, all of the instruments, or even an F grade in the course.*

Q1: Consider the following structure definitions:

```
struct student
{
      string reg_no;
      name student_name;
      int marks[5];
      float GPA;
};
```

```
struct name
{
  char  F Name[20];
  char L_Name[20];
};
```

Ask the user to enter number of students and then create a dynamic array of students. Read all of the above data for all the students except GPA. For each student calculate GPA according to his/her average marks (considering the grading scheme mentioned below).

For example :A student who obtained marks in five subjects as follows "60, 80, 90, 50, 60" will result in average marks of 68 and therefore his/her GPA
will be "2.87"

GPA	Marks (Percentage)
4.00	90-100
3.62	80-89
3.10	70-79
2.87	60-69
1.80	50-59
0.00	Below 50

(1) After calculating GPA for each student, display first names (*f_name*) and registration number (*reg_no*) of all those students whose GPA is above 3.0

(2) Sort data of all students (for example using bubble sort) so that students getting a higher GPA are stored first as compared to the students getting a lower GPA.

https://www.techopedia.com/definition/3757/bubble-sort

Q2: Consider the following structure definitions:

```
struct Employee
{
  char name[20];
  int scale;
  Salary YearlySalary[3];
  Tax YearlyTax[3];
  Address addr;
  Employee* next;
};
```

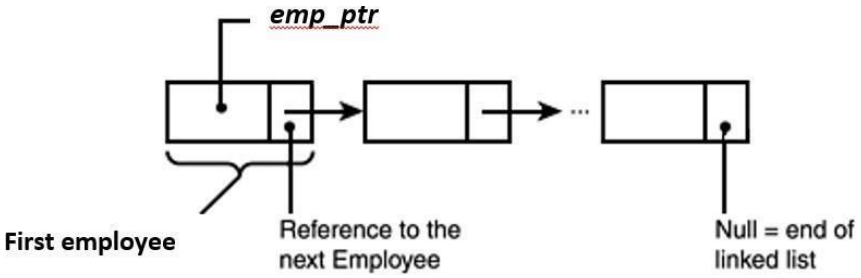
```
struct Address
{
  char city[20];
  int postcode;
  char email_id[20];
};
```

```
struct Salary
{
  int Year;
  float gross_salary;
};
```

```
struct Tax
{
  int Year;
  float tax_due;
  float net_salary;
};
```

Write a C++ program that asks the user to enter data for 5 employees. The following information should be entered by the user: *name*, *scale* (1 to 17), gross *salary* for the last 3 years, and *address*. The program should calculate the yearly tax (5% of gross salary for scale 1 to 10 employees and 7.5% for scale 11 to 17) and net_salary (gross_salary – tax_due).

For an employee, the structure variable should be allocated ***dynamically on heap memory*** using **new** operator. A pointer ***emp_ptr*** (of employee type) should be used to point to the first employee (initially ***emp_ptr*** will be NULL). The second employee should be referred by the first employee (**Employee* next** structure member). This will result in a chain of records in the heap (linked with each other using **next** pointer) as shown below.



Next, find the "average" net_salary for all the 10 employees. Also, display the name and address of the employee who has paid the highest tax (for any year).

Q3: Write a program that stores the following data about a Soccer player in a structure:

```
struct SoccerPlayer
 { char PlayerName[20];
     int Player_Number;
     int Points_Scored;
};
```

The program should keep a dynamic array of these structures. Each element is for a different player of the team. When the program runs it should display a menu asking the user to enter the data. The menu should provide the following options:

I) Add new players' information: *It will add information of a new player.*

2) Insert a new player: *It will insert information at the desired location.(for example at index 2).*

3) Delete the player's information *(Example: Delete player at index = 5)*

4) Display Player information: *Display information in a tabular format (for all players)*

5) Display: *The number and name of the player who has earned the most points should also be displayed.*

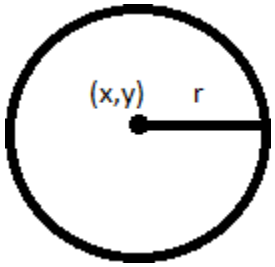
Input Validation: Do not accept negative values for players' numbers or points scored.

Q 4: Vehicle Class – A class called Vehicle is required by a programmer who is writing software for a car dealer. An object of the class Vehicle will consist of a registration number, the make of the vehicle, the year of manufacture and the current financial value of the vehicle. The first three of these will need to be set only at the time an object is created. The current value will also be set at the time of creation, butmay need to be changed during the vehicle's lifetime. You will need to write three files (Vehicle.h, Vehicle.cpp and Q1.cpp)

Your implemented class must fully provide the definitions of following class (interface) functions. Pleasealso 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 Vehicle{
  private:
    // think about the private data members...
  public:
    // provide definitions of following functions...

    Vehicle();// default constructor
    Vehicle(char *str);
    Vehicle(double);
    Vehicle(const Vehicle &);

    //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
    addVehicle();//adds a new vehicle
    ageOfVehicle();//returns age of vehicle on the basis of year passed
    getVehicleDetails();//return detail of vehicle
    getVehicleDetailsAtIndex();//return detail of vehicle stored at certain
    index
    isMatching();//returns true if passed vehicle matches with it
    returnByMake();//returns the array of vehicles of specific make
    returnByValue();//returns the array of vehicles of specific value
    returnByYear();//returns the array of vehicles of specific year
    vehicleSold();//deletes the specific vehicle once its sold
    ~Vehicle();
};

int main(){
    /you to create N vehicles objects for the car dealer
}
```

Q 5: Circle Class – A circle is identified by its center coordinates and its radius, as shown in the figure below. In the main file create N circles and implement a circle class whose specification is given below.You will need to write three files (Circle.h, Circle.cpp and Q2.cpp)



Your implemented class must fully provide the definitions of following class (interface) functions. Pleasealso 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 Circle{
  private:
    // think about the private data members...
  public:
    // provide definitions of following
    functions...
    Circle();// default constructor
    Circle(int x, int y);
    Circle(int x, int y, int radius);
    //you have to implement the following functions
    // think about the parameters required and return type of the following
    functions
    setCenter();//set center of a circle
    setRadius();//set radius of a
    circle
    getArea();//prints area of a
    circle
    returnLargestCircle();//return the largest circle from the array of circles
    overlapping();//determines of two circles are overlapping or not
    overlappingCircles();//returns an array of circles overlapping the largest
    circle
    ~Circle();
};

int main(){
    /you to create N circle objects and give user options to initialize them
}
```

Q 6: Scheduler Class – In order to generate schedule of a project, key information is of tasks/activities to do in that project. You have to implement a class which takes the task information from the user andgenerate different information related to schedule of the project. You will need to write three files (Scheduler.h, Scheduler.cpp and Q3.cpp). The program should first ask the total number of tasks in the project.

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.

You will have to learn Critical Path Method here: https://www.workamajig.com/blog/critical-path-method

---

```
Struct
  task{
  int id;
  int dur;
  int s_Time; //start time of each task
  int e_Time; //end time of each task
  int* dep; /*list of predecessors of this task - To simplify we assume that a
  highernumber task will depend on a lower number task e.g. T2 can depend on T1
  OR T4 can depend on T2 but the opposite is not true.*/

};

Class Scheduler{
  private:
    // think about the private data members...
  public:
    // provide definitions of following
    functions...
    Scheduler();// default constructor
    Scheduler(task* ts, int n);//initialized the project with n tasks

    //you have to implement the following functions
    // think about the parameters required and return type of the following
    functions

    setTaskDuration();//change task duration of all tasks
    set_nth_TaskDuration();//change duration of a specific task
    printTaskDependencyList();//print dependencies of a specific task
    completionTime();//print completion time of the project
    printCriticalTasks();/*returns array of critical tasks and displays them –
    sum of their duration should be equal to project completion time*/

    ~Scheduler();//destructor
}
```

Q 7: Implementation of Integer Class – Your goal is to implement “Integer" class. You will need to write three files (Integer.h, Integer.cpp and Q4.cpp). Your implemented class must fully provide the definitionsof 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 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 theresult
    static int numberOfLeadingZeros(int i); /*Returns the number of zerobits
    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 ofi in
    base 8
};
```
