Object Oriented Programming Lab Spring 2019

Aggregation and Composition

Lab 11

Submission Instructions

- ***Copy exact prototype from submission.cpp.***
- ***Submit all header files and cpp files.***

Task 1

Write a class **Point** that has the following data members.

- **X_Coordinate**: x coordinate of type integer
- **Y_Cooridnate**: y coordinate of type integer

The Point class has following member functions.

1. A default constructor that initializes the data members to zero.
   Point()
2. A parameterized constructor that accepts the parameters for each member variable.
   Point(int , int)
3. A copy constructor that takes a previously constructed object as an argument.
   Point(const Point &p)
4. Write accessors for each data member.
   int getX_Coordinate() **const**
   int getY_Coordinate() **const**

Write a class **Line** that represents a line segment between two Points hence it composes **Point** class. The Line class has the following data members.

Point_1: a point P1 of type Point
Point_2: a point P2 of type Point

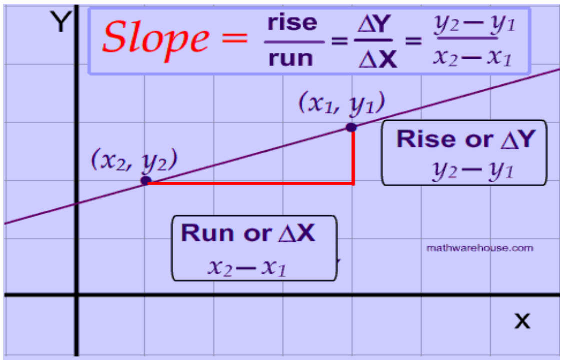
The Line class has the following member functions.
Note*: Use member initializer list for all constructors.

1. A default constructor that initializes the coordinates of 2 points to 4,6 and 2, 4.
   Line()

2. A parameterized constructor.
   Line(**int** x1, **int** y1, **int** x2, **int** y2)

3. A copy constructor that takes two previously constructed Point objects as argument.
   Line(**const** Point &p1, **const** Point &p2)

4. A member function findSlope that returns the slope of the length.
   float findSlope()


Y
$Slope = \frac{\text{rise}}{\text{run}} = \frac{\Delta Y}{\Delta X} = \frac{y_2 - y_1}{x_2 - x_1}$
(x_1, y_1)
(x_2, y_2)
Rise or ΔY
$y_2 - y_1$
Run or ΔX
$x_2 - x_1$
mathwarehouse.com
X

5. A member function findLength that returns the length of the line segment using distance formula.

   float findLength()

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

6. A member function findMidPoint that returns the midpoint of the line segment.
   In this function you have to create a pointer of type Point, dynamically allocate memory to the pointer, set midpoints and return it.

   Point& findMidPoint()

Midpoint Formula:

The midpoint (x,y) of a segment with endpoints (x_1, y_1) and (x_2, y_2) has coordinates

$$(x, y) = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)$$

Task 2

You are hired by a Garage owner to design a Garage Management System (GMS). To avoid the Traffic jams, your GMS will tell the new customers coming to the garage at the garage entrance that how many places are left in the Garage and whether there is place for their car in the Garage or not (Your GMS will also be conveying this information to Garage owner). To design this Garage Management System, you will write two classes:
Car and **Garage** class. Details of the functionalities desired by these classes shall be derived from the GMS requirements.

Write a class named **Car** having following data members.

- **make** of type string
- **carModel** of type string
- **regNo** of type string
- **year** of type int

Car Class will provide following interface.

1. Write a default constructor.
   Car()

2. Write parametrized constructor to initialize the attributes of car.
   Car(string m, string c, string r, string y)

3. Write getters and setters for car class.

Now write class **Garage** of your GMS having the following attributes

- **name** of type string

- **index** of type int that holds the value of occupied slots.

- **capacity** of type int

- Array of objects of **Car** class with maximum size equals to **capacity.**

Provide following functionality/interface for **Garage** class

1. Write a **default constructor** for class **Garage** to initialize its attributes. Initially consider that the Garage has capacity of 10 cars.
   Garage()

2. Write a **parametrized constructor** for class **Garage** to initialize its attributes.
   Garage(string n, int i, int c)

3. **bool IsEmpty** (): Returns true if garage is empty and false otherwise.

4. **bool IsFull** (): Returns true if garage is full and false otherwise.

5. **void Push** (car c): Implement this function to park a new car c in garage, considering that the garage is not full.

6. **bool Find** (string reg): Function should return true if the car is parked in garage and false otherwise.

7. **bool Remove** (string reg): Function should remove the car object from garage having reg_no equals to the one given in the parameter. Returns true if car is successfully removed and false otherwise and display messages accordingly.

8. **void Display** (): Displays all cars parked in garage currently.

Task 3

A university owns various departments (e.g., CS, Electrical Engineering), and each department has number of professors. If the university closes, the departments will no longer exist, but the professors in those departments will continue to exist. Therefore, a University can be seen as a composition of departments, whereas departments have an aggregation of professors. In addition, a Professor could work in more than one department, but a department could not be part of more than one university.

Write a Class named **Professor** having following attributes:

- **name** of type string
- **employeeID** of type int
- **designation** of type string

Write a Class named **Department** having following attributes:

- **name** of type string
- **deptID** of type int
- Array **profList** of type Professor
- **noOfProfessors** of type int

Write a Class named **University** having following attributes:

- **name** of type **string**
- Array **dept** of type **Department**
- **numberOfDepartments** of type **int**

Write following functions.

1. Write appropriate getter setter of each data member for each Class.

2. Add/delete/update Department in **University** class
   bool addDepartment(Department D)

bool deleteDepartment(string name)
bool updateDepartment(int id, string name) //Update name of department given department id.
void Display() function to display university information. Also display department information in this function.

3. Add/delete/update Professor in **Department** class
   bool addProfessor(Professor p)
   bool deleteProfessor (int id)
   bool updateProfessor (int id, string newDesignation) //Update designation of the professor given employee id
   void Display() function to display department information. Also display professors information in this function.