The objective of this lab is to:

Understand and practice composition and aggregation concepts of OOP.

Instructions!

1. Please follow the dress code before coming to the lab. Keep your student identity cards with you.
2. This is an individual lab, you are strictly **NOT** allowed to discuss your solutions with your fellow colleagues, even not allowed asking how is he/she is doing, it may result in negative marking. You can **ONLY** discuss with your TAs or with me.
1. Strictly follow good coding conventions (commenting, meaningful variable and functions names, properly indented and modular code.
2. Save your work frequently. Make a habit of pressing **CTRL+S** after every line of code you write.
3. Beware of memory leaks and dangling pointers.
4. Create separate class header file (.h) and class definition file (.cpp) for each task.

Task 01: [10 Marks]

The Westfield Carpet Company has asked you to write an application that calculates the price of carpeting for rectangular rooms. To calculate the price, you multiply the area of the floor (width times length) by the price per square foot of carpet. For example, the area of floor that is 12 feet long and 10 feet wide is 120 square feet. To cover that floor with carpet that costs $8 per square foot would cost $960. (12 × 10 ×8 = 960.)

```
// class FeetInches.

   class FeetInches
   {
          int feet;
          int inches;

   public:
          FeetInches(int f=0, int i=0)
          {
              feet = f;
              inches = i;
          }
          void setFeet(int val)
          {
               feet = val;
          }
          void setInches(int val)
          {
               inches = val;
          }
          int getFeet() const
          {
               return feet;
          }
          int getInches() const
          {
               return inches;
          }
   };
```

```
// class RoomDimension.

class RoomDimension
{
       FeetInches length;
       FeetInches Width;

public:
       // Constructor and other
       supported functions.

       FeetInches
       calculateArea();

};


// class RoomCarpet.

class RoomCarpet
{
       RoomDimension length;
       float
       cost_per_square_foot;

public:
       // Constructor and other
       supported functions.

       float
       calculateTotalCost();

};
```

First, you should create a class named **RoomDimension** that has two **FeetInches** (this class is given below) objects as attributes: one for the length of the room and one for the width. (you can overload multiply or other

operators in FeetInches class if required). The RoomDimension class should have a member function that returns the area of the room as a FeetInches object.

Next, you should create a **RoomCarpet** class that has a RoomDimension object as an attribute. It should also have an attribute for the cost of the carpet per square foot. The RoomCarpet class should have a member function that returns the total cost of the carpet.

Once you have written these classes, write menue driven code in ***main( )*** function to see the cost of carpeting in a room or to see the cost of carpeting in full house. If the user wants to see the cost of carpeting in a room then ask the user to enter the dimensions of a room and the price per square foot of the desired carpeting, your program should then display the total cost of the carpet in the given room. If the user wants to see the cost of carpeting in full house then ask the user to enter the number of rooms in house. Create a dynamic array of the size equal to the number of rooms in the house to store the RoomCarpet of each room. Then, ask the dimensions of each room in the house and cost per square from the user and populate the array with user-given data. Your program should then display total cost of the carpeting in the house.

Task 02: [10 Marks]

Game Studio is planning to develop a multiple player game for their next project. Each player will have its specific location on the screen, and can be moved in all four directions left,right, up and down. John is assigned the task to write a module to record the location of each player.

Help John to implement a **Location** Class that can ultimately be used to record the location of each player.All John knows is that Location class will have X and Y coordinates to record location and **Player** class to represent a player.

Task 03: [10 Marks]

FCIT has a number of Professors that are teaching in multiple departments. Management wants to have records of all the Professors that belong to the department of Software Engineering. Implement a Professor class that records the details of a Professor (name, designation and extension no), and a SEFaculty class that will have all the Professors of the specified department. Add a function ***displayFaculyMemebrs( )*** in SEFaculty class to display the list of all the professors in Department of SE.