Objective:

- It will help you understand the basic 'class' keyword.
- The private/public and information hiding.

Task-1: Time

The Time ADT, which we discussed today in class. Remember to make data members private.

Members:

```
int hour;
int minute;
int second;
```

1.1. void setHour (int h);
1.2. void setMinute (int m);
1.3. void setSecond (int s);
1.4. void setTime (int h, int m, int s);
1.5. int getHour ( );
1.6. int getMinute ( );
1.7. int getSecond ( );
1.8. void printTwentyFourHourFormat(); // print universal time
1.9. void printTwelveHourFormat(); // print standard time
1.10. void incSec(int = 1); // increment in the second of the calling time object
// default increment is 1
1.11. void incMin(int = 1); // increment in the minute of the calling time object
// default increment is 1
1.12. void incHour(int = 1); // increment in the hour of the calling time object
// default increment is 1

Input Validation: All Time object should have values of hour >= 1 and <= 23. And minute and second in range >= 1 and <= 59

Task-2: Rectangle

Create a class named as 'Rectangle' with attributes length and width. Provide member functions that calculate the perimeter and the area of the rectangle. Also, provide set and get functions for the length and width attributes.
Also, provide a predicate function *isSquare* that determines whether the rectangle is a square or not.

Setter function should make sure that length and width of rectangle must be greater than zero.

Task-3: Employee

Create a class called **Employee** that includes three pieces of information as data members. A first name (type char[100]), a last name (type char[100]) and a monthly salary (type float).

- Provide a set and a get function for each data member. Monthly Salary should be >= 0.
- Write a member function that increases an employee's monthly salary by 10 percent.
- Write a test program that demonstrates class Employee's functionalities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10 percent raise and display each Employee's yearly salary again.

Task-4: Student

Implement a class 'Student, whose object will be responsible for storing basic student information.

```
class Student
{
        char rollNo[11];        //roll no of student like BSEF14M001
        int semester;           //semester number
        char name[100];         //name of student
        float CGPA;
};
```

Add the following public functions in the Student class:

Getter/setter: you should know what to do with the setter/getter

```
void setRollNo(char *);
void setSemester( int );
void setName(char *);
void setCGPA(float);

const char * getRollNo();
int getSemester();
const char * getName()
float getCGPA();


bool isStudentDropOut(); //You get dropout if CGPA<1.5 in 1st semester
                         //You get dropout if CGPA<1.7 in 2nd and onward semester.

bool isStudentOnProbation(); //You get probation if CGPA>=1.5 and CGPA<2 in 1st semester
                             //You get probation if CGPA>=1.7 and CGPA<2in 2nd and onward
                             //semester.
```

Note: Make sure that the integral part of rollno must be >=1 and <=999;

Task-5: Car

Write a class named 'Car' that has the following member variables:

- ***yearModel***: An int that holds the car's year model.
- ***make***: A char array that holds the make of the car. E.g. Toyota etc.
- ***speed***: An int that holds the car's current speed.
- ***maxSpeed****:* The car maximum possible speed.

In addition, the class should have the following member functions.

- ***Accessor Methods***: Appropriate accessor functions to get the values stored in an object's yearModel, make, maxSpeed, and speed member variables.
- ***accelerate***: The accelerate function should add 5 to the speed member variable each time it is called.
- ***brake***: The brake function should subtract 5 from the speed member variable each time it is called.

Demonstrate the class in a program that creates a Car object, and then calls the accelerate function five times. After each call to the accelerate function, get the current speed of the car and display it. Then, call the brake function five times. After each call to the brake function, get the current speed of the car and display it.

Note: while brake, the current speed can't be less than zero and while accelerate, current car speed cant go above max speed.