UE19CS353 : OBJECT ORIENTED ANALYSIS AND DESIGN

Name : Priya Mohata

Section : E

SRN : PES2UG19CS301

Lab Work- Week-1 :

Q1. WAP to demonstrate the working of the access specifier(4).

```
class DemoAccess{
        private int age=21; // Private variable
        String srn; // Default variable
        public int marks; // public variable

        // Protected method
        protected void display()
        {
            System.out.println("Marks are:"+marks);
        }
}

class DemoProtect extends DemoAccess{
    Run | Debug
    public static void main(String[] args)
    {
        DemoProtect obj1=new DemoProtect();

        // Demonstrating public access modifier
        obj1.marks=98;

        //System.out.println("Age:"+obj1.age);
        System.out.println("Marks:"+obj1.marks);

        // Demonstrating default access modifier
        obj1.srn="PES2UG19CS301";
        System.out.println("SRN:"+obj1.srn);

        // Demonstrating protected access modifier
        obj1.display();
    }
}
```

O/P :

```
apple@Apples-MacBook-Air Week1 % javac program1.java
program1.java:16: error: age has private access in DemoAccess
        System.out.println("Age:"+obj1.age);
                                   ^
1 error
apple@Apples-MacBook-Air Week1 % javac program1.java
apple@Apples-MacBook-Air Week1 % java DemoProtect
Marks:98
SRN:PES2UG19CS301
Marks are:98
apple@Apples-MacBook-Air Week1 %
```

Q.2 Write a class named Car that has the following fields:----draw a class diagram

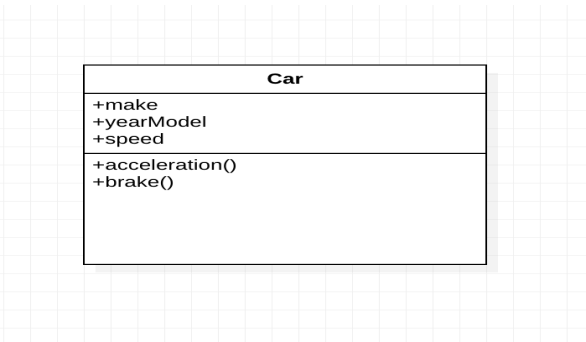
- **yearModel**. The yearModel is an int that holds the car's year model. For example, 2010.
- **make**. The make field references a String object that holds the make of the car. For example, Ford.
- **speed**. The speed field is an int that holds the car's current speed.

In addition the class should have the following

- Appropriate accessor methods should get the values stored in an object's yearModel, make, and speed fields.
- accelerate. The accelerate method should add 5 to the speed field each time it is called.
- brake. The brake method should subtract 5 from the speed field each time it is called.

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

Class Diagram :



Program :

```
class Car{
        int yearModel;
        String make;
        int speed;

        int limit_accelerate=5;
        int limit_brake=5;

        void acceleration()
        {
            speed=speed+5;
            limit_accelerate--;
            if(limit_accelerate<0){
                System.out.println("You have reached the limit!");
            }
            else
            {
                System.out.println("Speed is:"+speed);
            }
        }

        void brake()
        {
            speed=speed-5;
            limit_brake--;
            if(limit_brake<0){
                System.out.println("You have reached the limit!");
            }
            else
            {
                System.out.println("Speed is:"+speed);
            }
        }
}
```

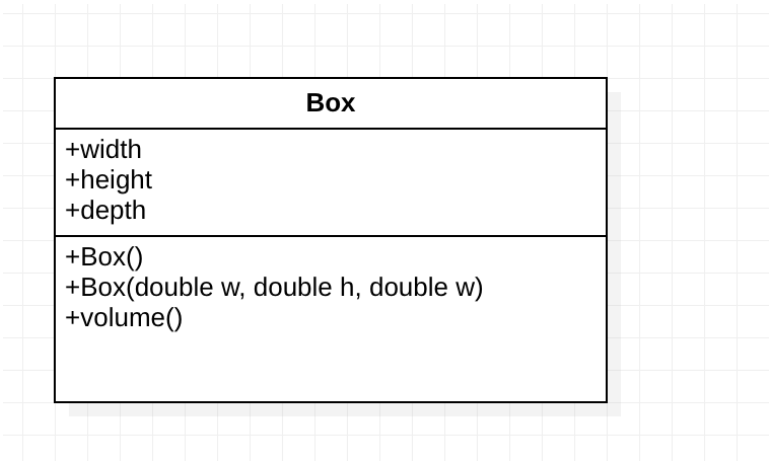
```
class Sample{
    Run | Debug
    public static void main(String args[]){
        // Creation of the object
        Car car=new Car();
        car.yearModel=2021;
        car.make="Volvo";
        car.speed=80;

        // Calling function 3 times first

        int count=0;
        System.out.println("-----Accelerating----");
        while(count<=5)
        {
        car.acceleration();
        count++;
        }

        System.out.println("-----Applying brake----");
        count=0;
        while(count<=5)
        {
            car.brake();
            count++;
        }
}
}
```

O/P:

```
apple@Apples-MacBook-Air Week1 % javac program2.java
apple@Apples-MacBook-Air Week1 % java Sample
-----Accelerating----
Speed is:85
Speed is:90
Speed is:95
Speed is:100
Speed is:105
You have reached the limit!
-----Applying brake----
Speed is:105
Speed is:100
Speed is:95
Speed is:90
Speed is:85
You have reached the limit!
apple@Apples-MacBook-Air Week1 %
```

Q3) Demonstrate the use of constructor by creating a Box class to compute volume.

Class Diagram :



Program :

```
class Box{
    double width;
    double height;
    double depth;

    // Default constructor - user defined

    Box(){
        width=0.0;
        height=0.0;
        depth=0.0;
    }

    // Parameterized constructor

    Box(double w,double h,double d){
        width=w;
        height=h;
        depth=d;
    }

    // Calculating the volume

    void volume(){
        System.out.println("The volume is :"+(width*height*depth));
    }

}
```

```java
class Main{
    Run | Debug
    public static void main(String[] args){

        // Initializing the default constructor
            Box b1=new Box();
            Box b2=new Box(10.0,20.0,15.0);

            b1.volume();

            b2.volume();
        // Initializing the parameterized constructor
    }
}
```

O/P :

```
apple@Apples-MacBook-Air Week1 % javac program3.java
apple@Apples-MacBook-Air Week1 % java Main
The volume is :0.0
The volume is :3000.0
apple@Apples-MacBook-Air Week1 %
```