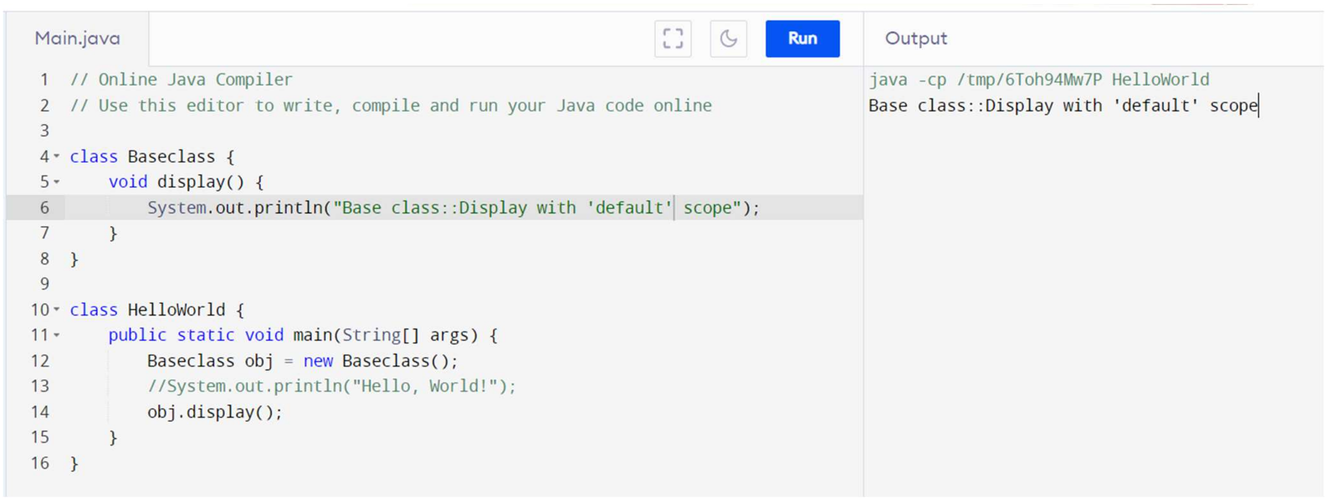
OOADJ lab

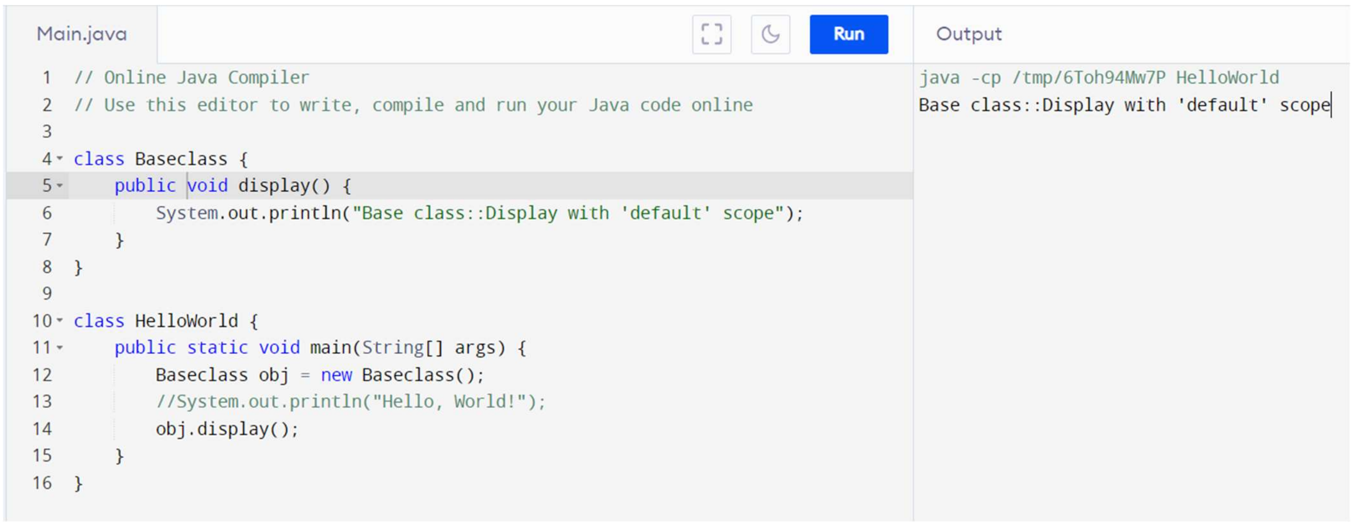
Name – B.Pravena Section - B
SRN – PES2UG19CS076

1. WAP to demonstrate access specifiers.



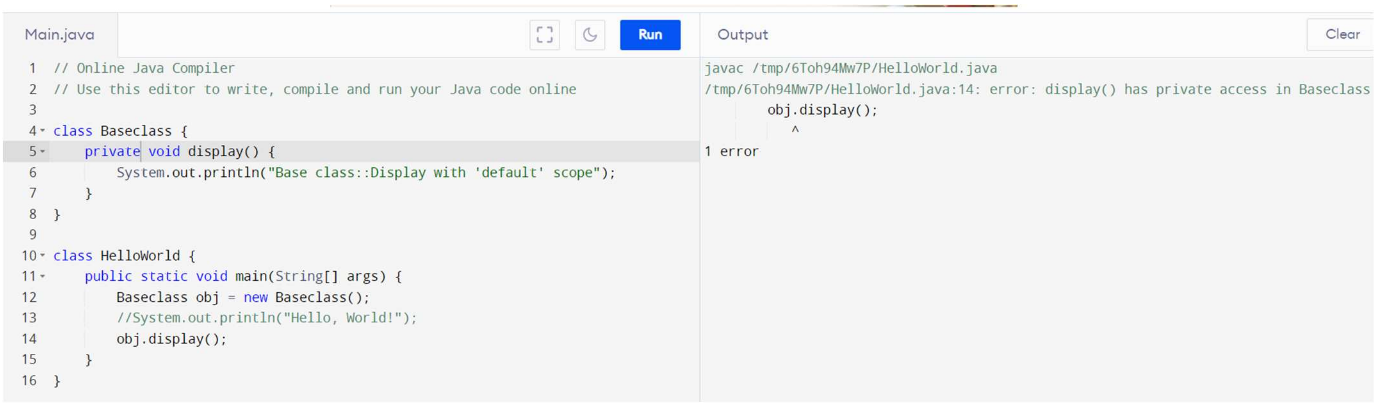
```java
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

class Baseclass {
    void display() {
        System.out.println("Base class::Display with 'default' scope");
    }
}

class HelloWorld {
    public static void main(String[] args) {
        Baseclass obj = new Baseclass();
        //System.out.println("Hello, World!");
        obj.display();
    }
}
```

```
java -cp /tmp/6Toh94Mw7P HelloWorld
Base class::Display with 'default' scope
```



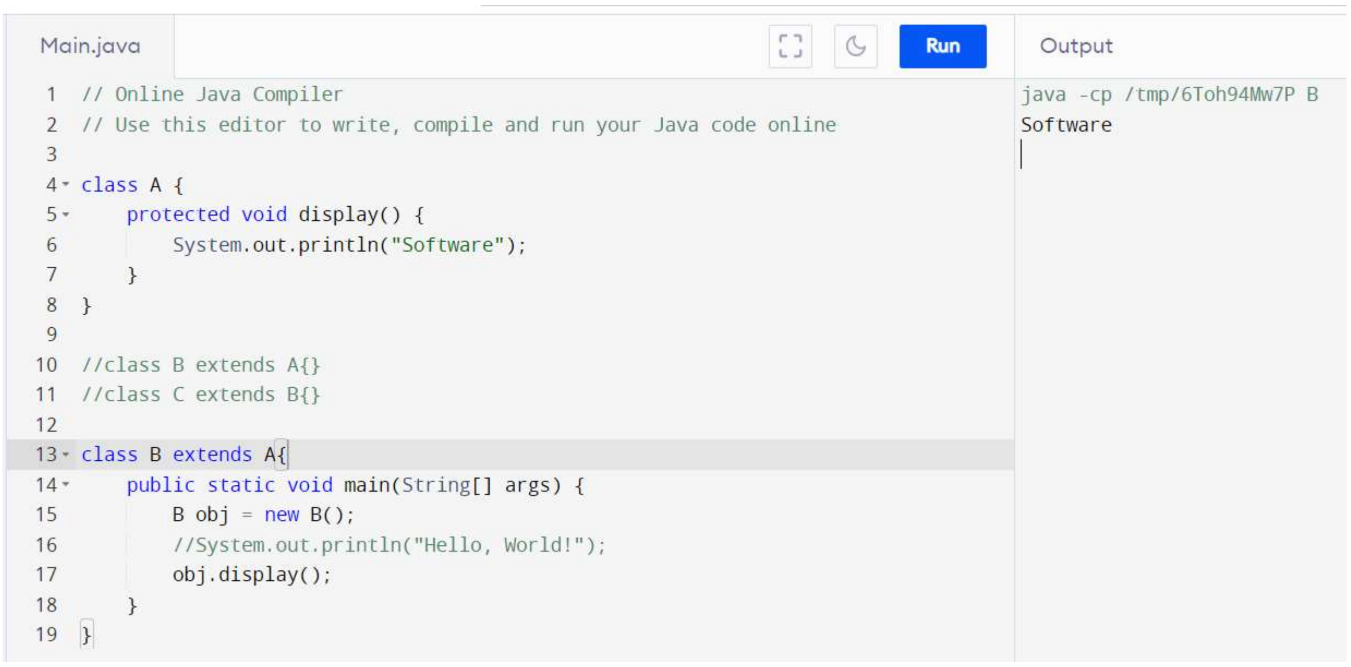
```java
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

class Baseclass {
    public void display() {
        System.out.println("Base class::Display with 'default' scope");
    }
}

class HelloWorld {
    public static void main(String[] args) {
        Baseclass obj = new Baseclass();
        //System.out.println("Hello, World!");
        obj.display();
    }
}
```

```
java -cp /tmp/6Toh94Mw7P HelloWorld
Base class::Display with 'default' scope
```



```java
1  // Online Java Compiler
2  // Use this editor to write, compile and run your Java code online
3
4  class Baseclass {
5      private void display() {
6          System.out.println("Base class::Display with 'default' scope");
7      }
8  }
9
10 class HelloWorld {
11     public static void main(String[] args) {
12         Baseclass obj = new Baseclass();
13         //System.out.println("Hello, World!");
14         obj.display();
15     }
16 }
```

```
javac /tmp/6Toh94Mw7P/HelloWorld.java
/tmp/6Toh94Mw7P/HelloWorld.java:14: error: display() has private access in Baseclass
        obj.display();
           ^
1 error
```



```java
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

class A {
    protected void display() {
        System.out.println("Software");
    }
}

//class B extends A{}
//class C extends B{}

class B extends A{
    public static void main(String[] args) {
        B obj = new B();
        //System.out.println("Hello, World!");
        obj.display();
    }
}
```

```
java -cp /tmp/6Toh94Mw7P B
Software
```

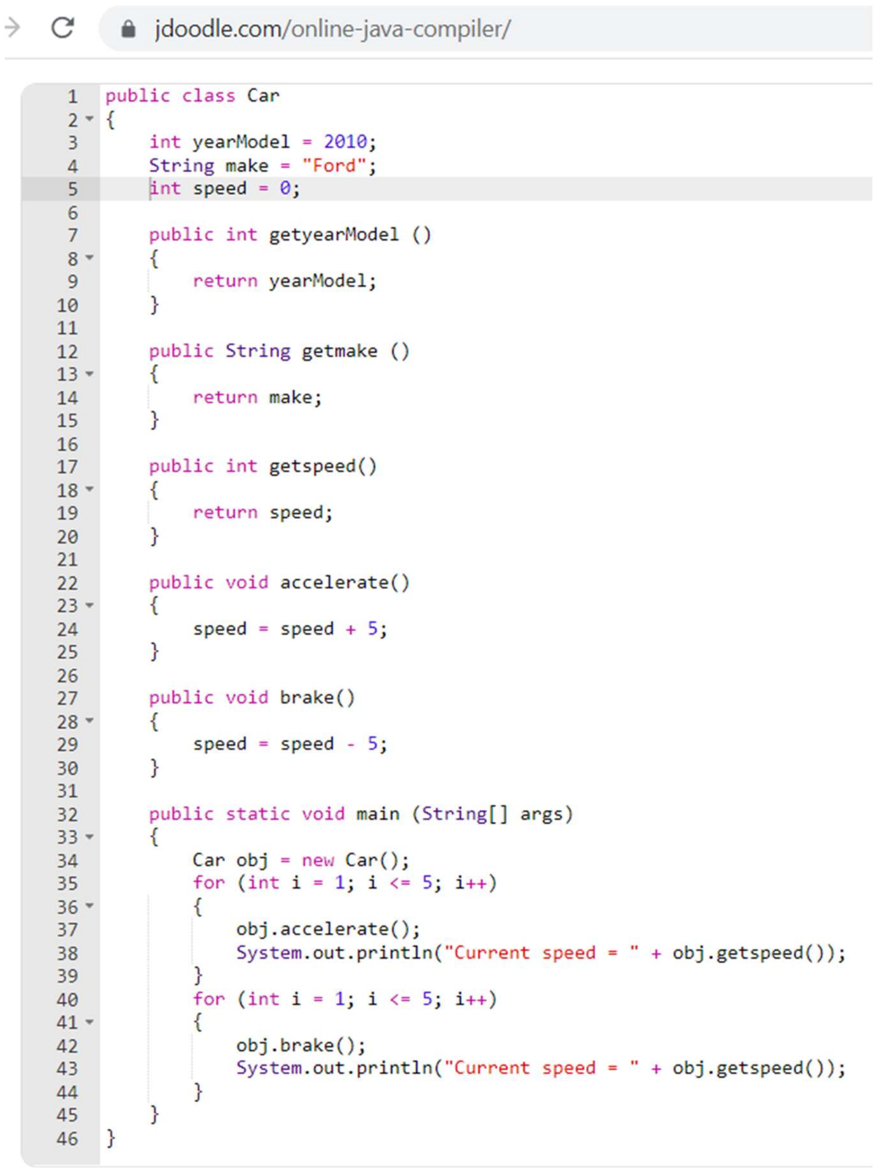
2) Write a class named Car that has the following fields:

- **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.

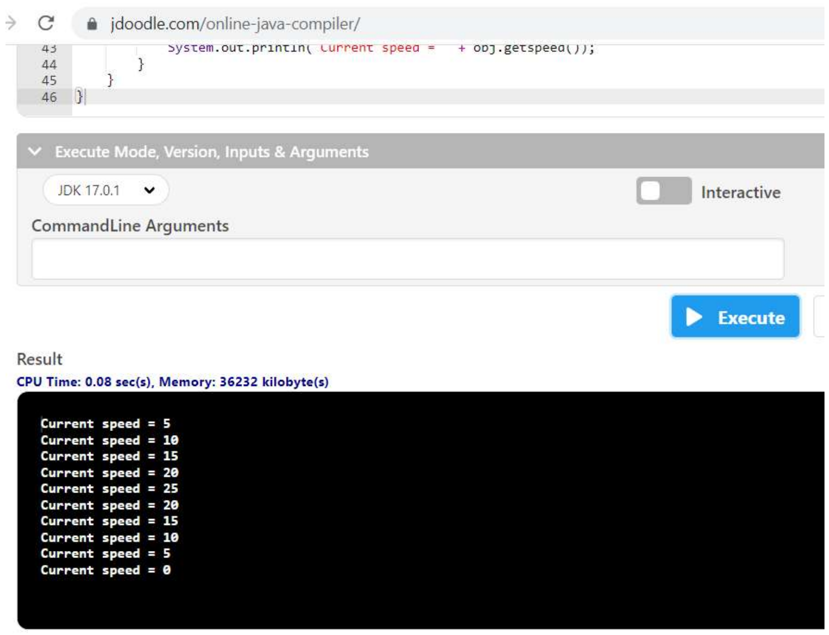
In addition the class should have the following

- Appropriate accessor methods should get the values stored in an object's yearModel, make, and speed fields.
- accelerate. The accelerate mthod 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.


jdoodle.com/online-java-compiler/

```java
public class Car
{
    int yearModel = 2010;
    String make = "Ford";
    int speed = 0;

    public int getyearModel ()
    {
        return yearModel;
    }

    public String getmake ()
    {
        return make;
    }

    public int getspeed()
    {
        return speed;
    }

    public void accelerate()
    {
        speed = speed + 5;
    }

    public void brake()
    {
        speed = speed - 5;
    }

    public static void main (String[] args)
    {
        Car obj = new Car();
        for (int i = 1; i <= 5; i++)
        {
            obj.accelerate();
            System.out.println("Current speed = " + obj.getspeed());
        }
        for (int i = 1; i <= 5; i++)
        {
            obj.brake();
            System.out.println("Current speed = " + obj.getspeed());
        }
    }
}
```


jdoodle.com/online-java-compiler/

```java
            System.out.println("Current speed = " + obj.getspeed());
        }
    }
}
```

Result

CPU Time: 0.08 sec(s), Memory: 36232 kilobyte(s)

```
Current speed = 5
Current speed = 10
Current speed = 15
Current speed = 20
Current speed = 25
Current speed = 20
Current speed = 15
Current speed = 10
Current speed = 5
Current speed = 0
```