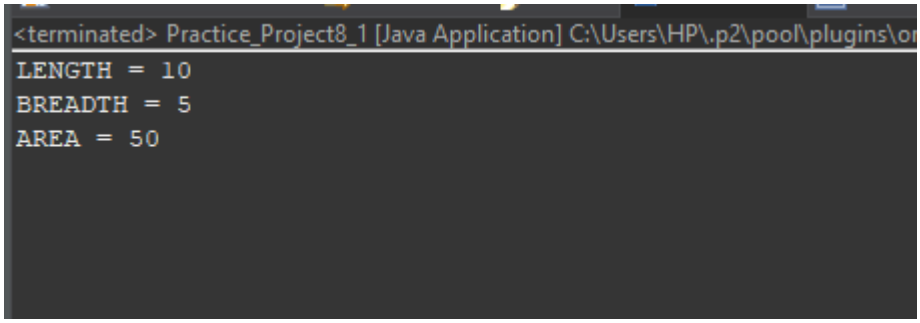
8. Write a program in Java to demonstrate the uses of classes, objects, and the object-oriented pillars in Java.

OBJECT AND CLASS

```
//implementing class, object
package assistedPracticeProject2;
public class Practice_Project8_1  //main class
{
	int length,breadth; //declaring variables
	Practice_Project8_1(int len,int bred) //parameterized constructor
	{	this.length=len;
		this.breadth=bred;   }
	void area() //method
	{
		System.out.println("LENGTH = "+length);
		System.out.println("BREADTH = "+breadth);
		System.out.println("AREA = "+(length * breadth));
	}
	public static void main(String[] args)
	{
		Practice_Project8_1 p1=new Practice_Project8_1(10,5); //class object
		p1.area(); //calling function area() using object
		//area();  //ERROR : method can only be accessed by object
	}
}
```

OUTPUT



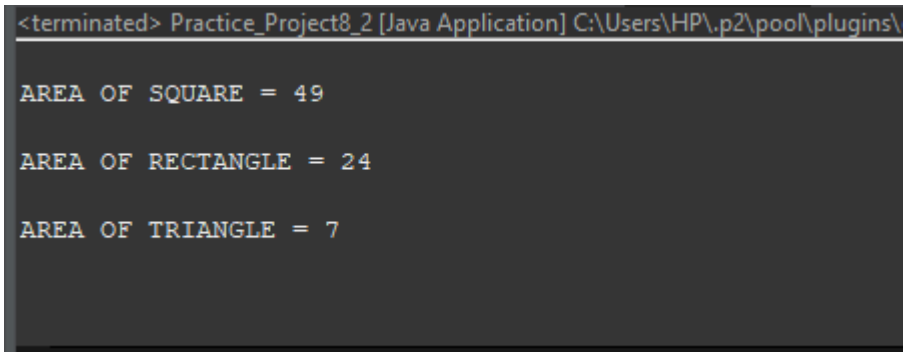
POLYMORPHISM

```
//implementing polymorphism
package assistedPracticeProject2;

public class Practice_Project8_2
{
	int length,breadth,side;
	double height,width;
	int result=0;
	void area(int side) //area() with single argument and integer return type
	{
		result=side*side; //calculating area of square
		System.out.println("\nAREA OF SQUARE = "+result);
	}
	void area(int length,int breadth) //area() with two arguments and integer return type
	{
		result=length*breadth; //calculating area of rectangle
		System.out.println("\nAREA OF RECTANGLE = "+result);
	}
	void area(double height,double width) //area() with two arguments and double return type
	{
		result=(int)(0.5*height*width); //calculating area of triangle
		System.out.println("\nAREA OF TRIANGLE = "+result);
	}
	public static void main(String[] args)
	{
		Practice_Project8_2 p1=new Practice_Project8_2();
		p1.area(7);
		p1.area(4,6);
		p1.area(4.5,3.2);
```

```
        }
}
```

OUTPUT



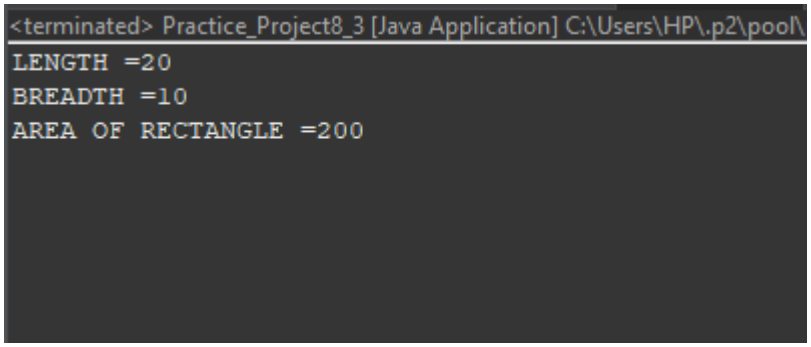
INHERITANCE

```
//implementing inheritance
package assistedPracticeProject2;

class area //super class
{
        protected int length=20; //length declaring as protected
}
public class Practice_Project8_3 extends area //subclass extending super class 'area'
{
        int breadth=10;
        public static void main(String[] args)
        {
                Practice_Project8_3 p1=new Practice_Project8_3();
                System.out.println("LENGTH ="+p1.length); //length is accessible to sub class
                System.out.println("BREADTH ="+p1.breadth);
                System.out.println("AREA OF RECTANGLE ="+(p1.length*p1.breadth));
```

```
	}
}
```

OUTPUT



ABSTRACTION

```
//implementing abstraction
package assistedPracticeProject2;

abstract class shape //abstract class
{
	int length=20;
	public abstract void area(); //abstract method
	public void display() //normal function
	{
		System.out.println("METHOD INSIDE ABSTRACT CLASS");
	}
}

class rectangle extends shape //sub class extending abstract class
{
	int breadth=4;
	public void area() //method
	{
```

```
                System.out.println("AREA = "+(length*breadth));


        }
}
public class Practice_Project8_4 {


        public static void main(String[] args)
        {
                rectangle r1=new rectangle(); //creating object for sub class
                r1.display();
                r1.area();
        }
}
```

OUTPUT

```
<terminated> Practice_Project8_4 [Java Application] C:\Users\HP\.p2\pool\plugi
METHOD INSIDE ABSTRACT CLASS
AREA = 80
```

ENCAPSULATION

```
//implementing encapsulation


package assistedPracticeProject2;
class employee
{
        private String empname; //private variable
        public void read(String ename)
        {
```

```
            this.empname=ename; //read name
        }
        public String disp()
        {
            return empname; //display name
        }
}

 class Practice_Project8_5
 {
        public static void main(String[] args)
        {
            employee e=new employee(); //creating object for employee class
            e.read("Manu"); //passing value
            System.out.println("NAME OF THE EMPLOYEE = "+e.disp()); //printing name
        }
}
```

OUTPUT