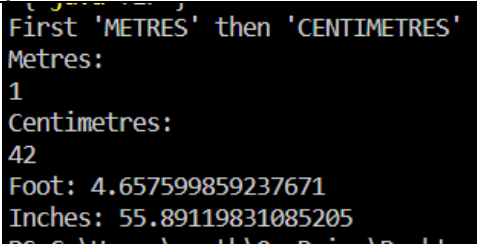
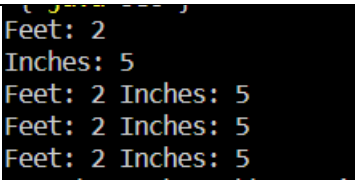


Module-2

Question 1:	Write a program to create a “distance” class with methods where distance is displayed in terms of feet and inches, how to create objects of a class.
Code:	<pre>import java.util.Scanner; public class distance_calc { public static void main(String[] args) { Distance d1 = new Distance(); double dist = d1.fun1(); System.out.println("Foot: " + dist * 3.28); System.out.println("Inches: " + dist * 3.28 * 12); } } class Distance { public double fun1() { Scanner sc = new Scanner(System.in); System.out.println("First 'METRES' then 'CENTIMETRES'"); System.out.println("Metres: "); int mtr = sc.nextInt(); System.out.println("Centimetres: "); int cms = sc.nextInt(); double res = mtr + ((float) cms / 100); return res; } // 12 inches is 1 foot };</pre>
Output:	 <pre>First 'METRES' then 'CENTIMETRES' Metres: 1 Centimetres: 42 Foot: 4.657599859237671 Inches: 55.89119831085205</pre>

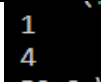
Question 2:	<p>Modify the “distance” class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.</p>
Code:	<pre>import java.util.Scanner; public class dist_refvar_clone { public static void main(String[] args){ Scanner sc = new Scanner(System.in); System.out.print("Feet: "); int feet = sc.nextInt(); System.out.print("Inches: "); int inches = sc.nextInt(); Distance2 dist = new Distance2(feet,inches) ; dist.leng(); Distance2 dist1 = new Distance2(dist.feet,dist.inches); dist1.leng(); Distance2 dist2 = new Distance2(dist1.feet,dist1.inches); dist2.leng(); } static class Distance2{ Scanner sc = new Scanner(System.in); int feet; int inches; Distance2(int feet,int inches){ this.feet = feet; this.inches = inches; } public void leng(){ System.out.println("Feet: "+this.feet + " Inches: "+this.inches); } } }</pre>
Output:	 <pre>{ java ... } Feet: 2 Inches: 5 Feet: 2 Inches: 5 Feet: 2 Inches: 5 Feet: 2 Inches: 5</pre>

Question 3:	Write a program in Java in which a subclass constructor invokes the constructor of the super class and instantiate the values.
Code:	<pre>import java.util.Scanner; public class callconstructor_with super { public static void main(String[] args){ Scanner sc = new Scanner(System.in); System.out.print("Enter Name: "); String a = sc.nextLine(); System.out.print("Enter age: "); int b = sc.nextInt(); char c = 'M'; System.out.print("Enter subject: "+c); System.out.println(); Teacher t1 = new Teacher(b,a,c); t1.emp_data(); t1.teac_data(); } static class Employee{ int age; String name; Employee(int a, String b){ this.age = a; this.name = b; } public void data(){ System.out.println("Name of Employee: "+name); System.out.println("Age of Employee: "+age); } } static class Teacher extends Employee{ char subj; Teacher(int a, String b,char c){ super(a,b); this.subj =c; } public void teacher_data(){ System.out.println("Name of Teacher: "+name); System.out.println("Age of Teacher: "+age); System.out.println("Subject of Teacher: "+subj); } } }</pre>

Output:	<pre> Enter Name: parth Enter age: 19 Enter subject: M Name of Employee: parth Age of Employee: 19 Name of Teacher: parth Age of Teacher: 19 Subject of Teacher: M </pre>
Question 4:	Write a program in Java to develop overloaded constructor. Also develop the copy constructor to create a new object with the state of the existing object.
Code:	<pre> import java.util.Scanner; public class overload_construct { public static void main(String[] args){ Scanner sc = new Scanner(System.in); System.out.print("Enter number for Factorial: "); int num = sc.nextInt(); System.out.print("Enter two numbers for Product: "); int num1 = sc.nextInt(); int num2 = sc.nextInt(); mulitConstructor mc = new mulitConstructor(num); mulitConstructor mc1 = new mulitConstructor(num1,num2); System.out.println("Factorial of given number is: "+mc.fact); System.out.println("Product of given number is: "+mc1.prod); } public static class mulitConstructor{ int fact; int prod; mulitConstructor(int num){ this.fact= fact(num); } mulitConstructor(int num,int num1){ this.prod = num*num1; } public static int fact(int n){ if(n<=1){ return 1; } else{ return n*fact(n-1); } } } } </pre>

Output:	<pre>/ { java } Enter number for Factorial: 4 Enter two numbers for Product: 3 6 Factorial of given number is: 24 Product of given number is: 18</pre>
---------	--------------------------------------------------------------------------------------------------------------------------------------------------------

Question 5:	Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
Code:	<pre>public class public_vs_public{ public static void main(String[]args){ reference r1 = new reference("Change"); try{ change(r1); } catch(Exception ob){ System.out.println(ob); } } public static void change(reference r){ // r = new reference("Change"); // System.out.println(r.PvtStr); System.out.println(r.FinalStr); System.out.println(r.PubStr); } } class reference{ String s1; private String PvtStr = "Hey there!"; public String PubStr = "Back again with movie recaps"; final String FinalStr = "It is what it is"; public reference(String Str){ s1 = Str; } }</pre>
Output:	<pre>It is what it is Back again with movie recaps Hey there!</pre>

Question 6:	Write a program to show the use of static functions and to pass variable length arguments in a function.
Code:	<pre>// import java.util.Scanner; public class staic_use { public static void main(String[] args){ // int n = sc.nextInt(); Varargs("A"); Varargs("B","C","D","E"); } public static void Varargs(String... s1){ System.out.println(s1); System.out.println(s1.length); } }</pre>
Output:	

Question 7:	Develop minimum 4 program based on variation in methods i.e., passing by value, passing by reference, returning values and returning objects from methods.
Code:	<pre> import java.util.Scanner; public class seventh { public static void main(String[] args) { String str = "JAVA"; Seventh s7 = new Seventh(); Seventh_1 s1 = new Seventh_1(); // pass by value // pass by reference // str = s7.str; Seventh_2 s2 = new Seventh_2(); Seventh_3 s3 = new Seventh_3(); System.out.println("This is pass by value: "); System.out.println("Before: " + s7.str); String str_1 = s7.str; System.out.println("After: " + str_1); System.out.println("This is pass by reference: "); System.out.println("Before: " + s1.strt); s1.strt = str; System.out.println("After: " + s1.strt); System.out.println("This is returning Values: "); System.out.println("Maximum between 420 and 69: " + s2.max); System.out.println("This is returning objects: "); s3.obj1(); s3.obj1(); } public static class Seventh { String str = "OOP"; } public static class Seventh_1 { String strt = "OOP"; } public static class Seventh_2 { int a = 420; int b = 69; int max = max(a, b); static int max(int a, int b) { if (a > b) { return a; } else { return b; } } } public static class Seventh_3 { Scanner sc = new Scanner(System.in); </pre>

	<pre>int a; int b; int min; void obj1() { this.a = sc.nextInt(); this.b = sc.nextInt(); this.min = min(this.a, this.b); System.out.println("Minimum among " + this.a + " & " + this.b + " : " + this.min); } static int min(int a, int b) { if (a < b) { return a; } else { return b; } } }</pre>
Output:	<pre>43 11 This is pass by value: Before: OOP After: OOP This is pass by reference: Before: OOP After: JAVA This is returning Values: Maximum between 420 and 69: 420 This is returning objects: 43 11 Minimum among 43 & 11 : 11 420 98 Minimum among 420 & 98 : 98</pre>

Question 8:	Write a program that implements two constructors in the class. We call the other constructor using 'this' pointer, from the default constructor of the class.
Code:	<pre>public class eight { public static void main(String[] args) { CallConstructor obj = new CallConstructor(); // System.out.println(obj); } static class CallConstructor { CallConstructor() { this("Start of Called Constructor", "Executing Commands in Called Constructor", "End of Default Constructor"); // System.out.println(); System.out.println("Command exactly after calling constructor from Default Constructor"); System.out.println("End of Default Constructor"); } CallConstructor(String st, String st1, String st2) { System.out.println(st); System.out.println(st1); System.out.println(st2); System.out.println(); } String name; } }</pre>
Output:	<pre>1) { java eight } Start of Called Constructor Executing Commands in Called Constructor End of Default Constructor Command exactly after calling constructor from Default Constructor End of Default Constructor PS C:\Users\parth\OneDrive\Desktop\DESKTOP\CODING\JavaScript\Java_Lab\La</pre>

Question 9:	Write a program in Java to demonstrate single inheritance, multilevel inheritance and hierarchical inheritance.
Code:	<pre>public class ninth { public static void main(String[] args){ a cl_ss = new a(); System.out.println("Calling method from object of 'a'"); cl_ss.print_a(); System.out.println(); b sing_inher = new b(); System.out.println("Calling method from object of 'b'"); sing_inher.print_b(); System.out.println(); c mult_inher = new c(); System.out.println("Calling method from object of 'c'"); mult_inher.print_c(); System.out.println(); d hier_inher = new d(); System.out.println("Calling method from object of 'd'"); hier_inher.print_d(); } public static class a{ static void print_a(){ System.out.println("Inside of 'a' class"); System.out.println("End of method of 'a' class"); } } public static class b extends a{ static void print_b(){ System.out.println("Inside 'b' class"); System.out.println("Calling method of parent class 'a'); print_a(); System.out.println("End of method of 'b' class"); } } }</pre>

```
    }  
    public static class c extends b{  
        static void print_c(){  
            System.out.println("Inside 'c' class");  
            System.out.println("Calling method of parent class  
'b'");  
            print_b();  
            System.out.println("Calling method of grandparent  
class 'a'");  
            print_a();  
            System.out.println("Ending method of 'c' class");  
        }  
    }  
    public static class d extends a{  
        static void print_d(){  
            System.out.println("Inside 'd' class");  
            System.out.println("Calling method of parent class 'a'  
");  
            print_a();  
            System.out.println("End of method 'd' class");  
        }  
    }  
}
```

Output:

```
17 { java main }  
Calling method from object of 'a'  
Inside of 'a' class  
End of method of 'a' class  
  
Calling method from object of 'b'  
Inside 'b' class  
Calling method of parent class 'a'  
Inside of 'a' class  
End of method of 'a' class  
End of method of 'b' class  
  
Calling method from object of 'c'  
Inside 'c' class  
Calling method of parent class 'b'  
Inside 'b' class  
Calling method of parent class 'a'  
Inside of 'a' class  
End of method of 'a' class  
End of method of 'b' class  
Calling method of grandparent class 'a'  
Inside of 'a' class  
End of method of 'a' class  
Ending method of 'c' class  
  
Calling method from object of 'd'  
Inside 'd' class  
Calling method of parent class 'a'  
Inside of 'a' class  
End of method of 'a' class  
End of method 'd' class
```

Question 10:	Java Program to demonstrate the real scenario (e.g., bank) of Java Method Overriding where three classes are overriding the method of a parent class. Creating a parent class.
Code:	<pre>// package Lab_9; import java.util.Scanner; public class over_ride{ public static void main(String[] args){ HDFC acc_1 = new HDFC(); System.out.println("IFSC code of A branch of HDFC bank is: "+acc_1.getIFSC("HDFC")); ICICI acc_2 = new ICICI(); System.out.println("IFSC code of B branch of ICICI bank is: "+acc_2.getIFSC("ICICI")); SBI acc_3 = new SBI(); System.out.println("IFSC code of C branch of SBI bank is: "+acc_3.getIFSC("SBI")); } public static class HDFC{ String getIFSC(String bnk){ Scanner sc = new Scanner(System.in); System.out.print("Enter the numeric part of the IFSC code: "); int ifs = sc.nextInt(); return bnk+ifs; } } public static class ICICI extends HDFC{}; public static class SBI extends HDFC{}; }</pre>
Output:	<pre> Enter the numeric part of the IFSC code: 65845 IFSC code of A branch of HDFC bank is: HDFC65845 Enter the numeric part of the IFSC code: 56213 IFSC code of B branch of ICICI bank is: ICICI56213 Enter the numeric part of the IFSC code: 987456 IFSC code of C branch of SBI bank is: SBI987456 </pre>

Question 11:	Write a program that implements simple example of Runtime Polymorphism with multilevel inheritance. (e.g., Animal or Shape)
Code:	<pre>public class runtime_poly { public static void main(String[] args) { Quadrilateral q1 = new Quadrilateral(); q1.getInfo("quadrilateral"); System.out.println(); Parallelogram p1 = new Parallelogram(); p1.getInfo("parallelogram"); p1.getExtra("parallelogram"); System.out.println(); Rhombus r1 = new Rhombus(); r1.getInfo("rhombus"); r1.getExtra("rhombus"); System.out.println(); Square s1 = new Square(); s1.getInfo("Square"); s1.getExtra("Square"); } public static class Quadrilateral { void getInfo(String shape) { System.out.println("No. of sides in " + shape + " is: 4"); } } public static class Parallelogram extends Quadrilateral { void getExtra(String shape) { System.out.println("Opposite Sides in " + shape + " have same length"); } } public static class Rhombus extends Parallelogram { void getExtra(String shape) { System.out.println("All sides in " + shape + " have same length"); } } public static class Square extends Rhombus {</pre>

	<pre>void getExtra(String shape) { System.out.println("All sides in " + shape + " have same length"); System.out.println("All angles in " + shape + " measure same i.e 90 degrees"); } }</pre>	
Output:	<pre>No. of sides in quadrilateral is: 4 No. of sides in parallelogram is: 4 Opposite Sides in parallelogram have same length No. of sides in rhombus is: 4 All sides in rhombus have same length No. of sides in Square is: 4 All sides in Square have same length All angles in Square measure same i.e 90 degrees</pre>	

Question 12:	Write a program to compute if one string is a rotation of another. For example, pit is rotation of tip as pit has same character as tip.
Code:	<pre>import java.util.Scanner; public class string_rot{ public static void main(String[] args){ Scanner sc = new Scanner(System.in); System.out.println("Enter the first string: "); String str = sc.nextLine(); System.out.println("Enter the first string: "); String str_1 = sc.nextLine(); int check=0; if(str.length()==str_1.length()){ for(int i=0;i<str.length();i++){ if(str.charAt(i)==str_1.charAt(str_1.length()-i-1)){ check++; } } else{ break; } } if(check!=str.length()){ System.out.println("Strings are not in rotation"); } else{ System.out.println("Strings are in rotation"); } } }</pre>

Output:	<pre>(p.) { java main } Enter the first string: manav Enter the first string: vanam Strings are in rotation or Strings are not in rotation</pre>	<pre>(p.) { java main } Enter the first string: manav Enter the first string: navam Strings are not in rotation or Strings are in rotation</pre>
---------	--------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------

Question 13:	Describe abstract class called Shape which has three subclasses say Triangle, Rectangle, Circle. Define one method area() in the abstract class and override this area() in these three subclasses to calculate for specific object i.e. area() of Triangle subclass should calculate area of triangle etc. Same for Rectangle and Circle.
Code:	<pre>import java.util.Scanner; public class thirteen{ public static void main(String[] args){ Scanner sc = new Scanner(System.in); System.out.println("Enter your choice: "); System.out.println("1: Area of Triangle "); System.out.println("2: Area of Rectangle "); System.out.println("3: Area of Circle "); System.out.println("4: Exit "); int choice = sc.nextInt(); switch(choice){ case 1: Triangle t1 = new Triangle(); System.out.println("Enter height of triangle: "); t1.b = sc.nextDouble(); System.out.println("Enter length of triangle: "); t1.h = sc.nextInt(); t1.fun(); // System.out.println("Area: "+t1.area); break; case 2: Rectangle r1 = new Rectangle(); System.out.println("Enter length of Rectangle: "); r1.l = sc.nextInt(); System.out.println("Enter breadth of Rectangle: "); r1.b = sc.nextInt(); r1.fun(); // System.out.println("Area: "+r1.area); break; case 3: Circle c1 = new Circle();</pre>

```
        System.out.println("Enter radius of Circle: ");
        c1.r = sc.nextDouble();
        c1.fun();
        //    System.out.println("Area: "+c1.area);
        break;
    case 4:
        System.out.println("Thank you for using!!");
        break;
    }

}

}

abstract class Shape{
    void area(int h,double b){
        System.out.println("In the triangle");
        System.out.println(0.5*h*b);
    }
    void area(int l, int b){
        System.out.println("In the Rectangle");
        System.out.println(l*b);
        //    return l*b;
    }
    void area(double r){
        //    return (3.14*r*r);
        System.out.println("In the Circle");
        System.out.println(3.14*r*r);
    }
}

class Triangle extends Shape{
    int h;
    double b;
    void fun(){
        area(this.h,this.b);
    }
}
```

	<pre> class Rectangle extends Shape{ int l; int b; void fun(){ area(this.l,this.b); } // int area = area(this.l,this.b); } class Circle extends Shape{ double r; void fun(){ area(this.r); } // double area = area(this.r); } </pre>		
Output:	<pre> Enter your choice: 1: Area of Triangle 2: Area of Rectangle 3: Area of Circle 4: Exit 1 Enter height of triangle: 12 Enter length of triangle: 3 In the triangle 18.0 </pre>	<pre> Enter your choice: 1: Area of Triangle 2: Area of Rectangle 3: Area of Circle 4: Exit 3 Enter radius of Circle: 4 In the Circle 50.24 </pre>	<pre> Enter your choice: 1: Area of Triangle 2: Area of Rectangle 3: Area of Circle 4: Exit 4 Thank you for using!! </pre>

Question 14:	Write a program in Java to demonstrate multiple inheritance
Code:	<pre>interface vehicleone{ int speed=90; public void distance(); } interface vehicletwo{ int distance=100; public void speed(); } class Vehicle implements vehicleone,vehicletwo{ public void distance(){ int distance=speed*100; System.out.println("distance travelled is "+distance); } public void speed(){ int speed=distance/100; } } class MultipleInheritanceUsingInterface{ public static void main(String args[]){ System.out.println("Vehicle"); obj.distance(); obj.speed(); } }</pre>

Output:

```
distance travelled is 9000
```


Question:15	<p>a) Write an application that illustrates method overriding in the same package and different packages.</p> <p>b) Also demonstrate accessibility rules in inside and outside packages.</p>
Code:	<pre>Package pack; Public class same_pack{ Public void msg(){ System.out.println("Hey there"); } Import pack.same_pack; Public class diff extends pack.same_pack{ Public void msg(){ System.out.println("Back there with movie recaps"); } Public static void main(String args[]) { Pack.same_pack=new pack.A(); Obj.msg(); Obj=new diff(); Obj.msg(); } }</pre>
Output:	<pre>Hey there Back there with movie recaps</pre>