1) Program using scanner class

```
import java.util.Scanner;
public class Scanner_Class {

public static void main(String[] args) {
int a;
String b;
float c;
System.out.println("Enter the value of A ==> ");
Scanner sc =new Scanner(System.in);
a=sc.nextInt();

System.out.println("Enter your name ==> ");
Scanner sc1 =new Scanner(System.in);
b=sc1.nextLine();

System.out.println("Enter Any decimal point ==> ");
Scanner sc2 =new Scanner(System.in);
c=sc2.nextFloat();
System.out.println("The value which you have entered is ==> "+a);
System.out.println("Your name is ==> "+b);
System.out.println("The Decimal value which you have entered is ==> "+c);
     }
}
```

Output -:

```
Enter the value of A ==>
5
Enter your name ==>
Shekhar
Enter Any decimal point ==>
4.4
The value which you have entered is ==> 5
Your name is ==> Shekhar
The Decimal value which you have entered is ==> 4.4
```

Program on unsigned right shift operator

```java
public class Shifting_Operator {

    public static void main(String[] args) {
    int x=10,y=-10,a,b,c,d;
    System.out.println("The binary representation of x = "+Integer.toBinaryString(x));
    System.out.println("The binary representation of y = "+Integer.toBinaryString(y));

    a=x>>2;
    System.out.println("a = "+a);
    System.out.println("The binary representation of a = "+Integer.toBinaryString(a));

    b=x>>>2;
    System.out.println("b = "+b);
    System.out.println("The binary representation of b = "+Integer.toBinaryString(b));

    c=x<<2;
    System.out.println("c = "+c);
    System.out.println("The binary representation of c = "+Integer.toBinaryString(c));

    d=y>>>2;
    System.out.println("d = "+d);
    System.out.println("The binary representation of d = "+Integer.toBinaryString(d));

    }

}
```

Output -:

```
The binary representation of x = 1010
The binary representation of y = 11111111111111111111111111110110
a = 2
The binary representation of a = 10
b = 2
The binary representation of b = 10
c = 40
The binary representation of c = 101000
d = 1073741821
The binary representation of d = 111111111111111111111111111101
```

2) While Loop

```
public class While {

	public static void main(String[] args) {
		int i=1;
		System.out.print("The even no. is ==> ");
		while (i<=20)
		{
			if (i%2==0)
			{
				System.out.print(i+" ");
			}
			i++;
		}
	}
}
```

Output -:

```
The even no. is ==> 2 4 6 8 10 12 14 16 18 20
```

// For Loop

```
public class For {
	public static void main(String[] args) {
		int i,j;
		for (i=0;i<5;i++)
		{
			for(j=0;j<=i;j++)
			{
				System.out.print(" * ");
			}
			System.out.println();
		}
	}
}
```

Output -:

```
	*
	*  *
	*  *  *
	*  *  *  *
	*  *  *  *  *
```

// Do While Loop

```
public class Do_While {
    public static void main(String[] args) {
        int i=0;
        System.out.print("The Odd no. are ==> ");
        do
        {
            if(i%2!=0)
            {
                System.out.print(i+" ");
            }
            i++;
        }while(i<=20);
    }
}
```

Output -:

```
The Odd no. are ==> 1 3 5 7 9 11 13 15 17 19
```

// Label Break

```
public class Break {

    public static void main(String[] args) {
        int j,k;
        outermost:
            for(j=1;j<5;j++)
            {
                inermost:
                    for(k=1;k<3;k++)
                    {
                        System.out.println("J = "+j+ " and K = "+k);
                    }
                if(j==3)
                break outermost;
            }
    }
}
```

Output -:

```
J = 1 and K = 1
J = 1 and K = 2
J = 2 and K = 1
J = 2 and K = 2
J = 3 and K = 1
J = 3 and K = 2
```

// Label Continue

```
public class Continue {

    public static void main(String[] args) {
        int x = 1;
        int y = 10;
        do {
            if(x==y/2)
            {
                x++;
                continue;
            }
            System.out.print(x+" ");
            x++;
        }while(x<=y);
    }
}
```

Output -:

```
1 2 3 4 6 7 8 9 10
```

3) Create Class

```
class Create{
    int roll;
    String name;
    void Data(int a, String b)
    {
        roll=a;
        name=b;
    }
    void display()
    {
        System.out.println(roll+" "+name);
    }
}
public class Create_Class {
    public static void main(String[] args) {
    Create s1=new Create();
    s1.Data(5,"Shekhar");
    s1.display();
    Create s2=new Create();
    s2.Data(45,"Jitendra");
    s2.display();
    }
}
```

Output -:

```
 5 Shekhar
45 Jitendra
```

4) Default Constructor

```java
class Default
{
        int rno;
        String name;
}
public class Default_Contructor {
        public static void main(String[] args) {
                Default obj = new Default();
                System.out.println(obj.rno);
                System.out.println(obj.name);
        }
}
```

Output -:

```
0
null
```

// Parameterized Constructor

```java
class Parameterized
{
        int number;
        String name;
        Parameterized(int a, String b)
        {
                number=a;
                name =b;
        }
        void display()
        {
                System.out.println(+number+ "\t " +name);
        }
}
public class Parameterized_Constructor {

        public static void main(String[] args) {
                Parameterized c1=new Parameterized(1947, "India");
                c1.display();
                Parameterized c2=new Parameterized(1960, "Maharashtra");
                c2.display();
        }
}
```

Output -:

```
1947    India
1960    Maharashtra
```

5) Method Overloading

```
class Method
{
      int a,b,c;
      float d,e;
      int add(int x,int y)
      {
            return(x+y);
      }
      float add(int x,float y)
      {
            return(x+y);
      }
      float add(float x,float y)
      {
            return(x+y);
      }
}
public class Method_Overloading {

      public static void main(String[] args) {
      Method obj1 =new Method();
      int s= obj1.add(4, 16);
      System.out.println("The addition is ==>"+s);
      float v=obj1.add(6, 16.4f);
      System.out.println("The addition is ==>"+v);
      float w=obj1.add(8.5f, 16.4f);
      System.out.println("The addition is ==>"+w);
      }
}
```

Output -:

```
The addition is ==>20
The addition is ==>22.4
The addition is ==>24.9
```

6) Passing Object as an argument to an method

```
public class Passing {
        int i=5;
        void increment(Passing ob1)
        {
                ob1.i++;
        }
        public static void main(String[] args) {
                Passing obj = new Passing();
                System.out.println(" Before increment a value ==> "+obj.i);
                obj.increment(obj);
                System.out.println(" After increment a value ==> "+obj.i);
        }
}
```

Output -:

```
 Before increment a value ==> 5
After increment a value ==> 6
```

// Returning object from a method

```
class Shirt{
        String colour;
        int size;
        Shirt(String a, int b)
        {
                colour = a;
                size = b;
        }
        Shirt doublesize(Shirt s1)
        {
                Shirt newshirt = new Shirt(s1.colour = "Yellow", s1.size*2);
                return newshirt;
        }
}
public class Return {
        public static void main(String[] args) {
                Shirt obj =new Shirt("Red",10);
                obj=obj.doublesize(obj);
                System.out.println(obj.colour);
                System.out.println(obj.size);
        }
}
```

Output -:

```
Yellow
 20
```

7) Program on public access modifier

```
//package ==> Define
// class  ==> Access_Modifier
// Use public,protected,private,default access modifier
package Define;
import User.User_Define_Package;    //To access Method and data in first package
                                          (i.e. User_Define_Package)

class Dept_info extends Access_Modifier{
      void display()
      {
            System.out.println("Lab name is = "+labname);
      }
}
public class Access_Modifier {
      protected String labname ="Language lab";
      private String name = "Sheetal Mam";
      static int id = 1234;   //Default modifier
      public static void main(String[] args) {
            User_Define_Package udp =new User_Define_Package();
            udp.teach();
            Dept_info di =new Dept_info ();
            di.display();
            Access_Modifier am =new Access_Modifier();
            System.out.println("Teachers name is ==> "+am.name);
            System.out.println("Teachers id is ==> "+am.id);
      }
}
```

Output -:

```
Teacher is teaching well
Lab name is = Language lab
Teachers name is ==> Sheetal Mam
Teachers id is ==> 1234
```

8) Program to print 1-D array

```
public class Array {

	public static void main(String[] args) {
		int a[] = {1,2,3};
		System.out.print("The elements are ");
		for(int i=0; i<3;i++)
		{
			System.out.print(a[i]+ " ");
		}
	}
}
```

Output -:

```
The elements are 1 2 3
```

// Program to print 1-D array by using Scanner

```
import java.util.Scanner;
public class ScannerArray {
	public static void main(String[] args) {
	int a[]=new int[100];
	System.out.println("Enter the size of array");
	Scanner sc =new Scanner(System.in);
	int n=sc.nextInt();
	System.out.println("Enter a array element");
	for (int i=0; i<n; i++)
	{
		a[i]=sc.nextInt();
	}
	System.out.print("The array elements is ==> ");
	for (int i=0; i<n; i++)
	{
		System.out.print(a[i]+" ");
	}
	}
}
```

Output -:

```
Enter the size of array
5
Enter a array element
1 2 3 4 5
The array elements is ==> 1 2 3 4 5
```

9) Print 2-D array

```
public class Array_TwoD {
        public static void main(String[] args) {
                int a[][]= {{1,2,3},{4,5,6},{7,8,9}};
                System.out.println("Array element is ");
                for(int i=0;i<3;i++)
                {
                        for(int k=0;k<3;k++)
                        {
                                System.out.print( a[i][k]+" ");
                        }
                System.out.println();
                }
        }
}
```

Output -:

```
Array element is
1 2 3
4 5 6
7 8 9
```

// Scan and Print 2-D array

```
public class Scan_Array {

        public static void main(String[] args) {
        int a[][]= {{1,2,3},{4,5,6},{7,8,9}};
        int b[][]= {{9,8,7},{6,5,4},{3,2,1}};
        int c[][]= {{7,5,3},{4,8,6},{8,3,6}};
        System.out.println("The array elements are");
        System.out.println("The matrix A is -: ");
        for(int i=0;i<3;i++)
        {
                for(int j=0;j<3;j++)
                {
                        System.out.print(a[i][j]+" ");
                }
                System.out.println();
        }
        System.out.println("The matrix B is -: ");
        for(int i=0;i<3;i++)
        {
                for(int j=0;j<3;j++)
                {
                        System.out.print(b[i][j]+" ");
                }
                System.out.println();
        }
        System.out.println("The matrix C is -: ");
```

```
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            System.out.print(c[i][j]+" ");
        }
        System.out.println();
    }
    }
}
```

Output -:

```
    The array elements are
The matrix A is -:
    1 2 3
    4 5 6
    7 8 9
The matrix B is -:
    9 8 7
    6 5 4
    3 2 1
The matrix C is -:
    7 5 3
    4 8 6
    8 3 6
```

10) Program on String

```
public class StringDemo {

      public static void main(String[] args) {
            String d = "Dhoni";
            d=d.concat(" is a cricketer" ); //concating both name (i.e Dhoni is a cricketer)
            System.out.println(d);
            String i = "Maharashtra";
            //Length of String i (i.e Maharashtra)
            System.out.println("The length of the string is = "+i.length());
            // Which character at 2nd index
            System.out.println("The character of 2nd index of the string is = "+d.charAt(2));
            // It is compare that what is the ascii value of these two strings
            System.out.println("Ascii value of both string is = "+i.compareTo(d));
            // It will show that both name is equal or not
            System.out.println("Is the both name are same = "+d.equals(i));
            // From 1st index whole name will be executed
            System.out.println("The substring is "+i.substring(1));
            //r is replace by s
            System.out.println("After replacing the name is = "+i.replace('r', 's'));

      }

}
```

Output -:

```
Dhoni is a cricketer
The length of the string is = 11
The character of 2nd index of the string is = o
Ascii value of both string is = 9
Is the both name are same = false
The substring is aharashtra
After replacing the name is = Mahasashtsa
```

11)Program on string buffer

```
public class StringBufferExample {

	public static void main(String[] args) {
	StringBuffer sb= new StringBuffer("Hello");
	sb.append("Java");
	System.out.println(sb);
	StringBuffer sb1= new StringBuffer("Hello");
	sb1.replace(1,2,"Java");
	System.out.println(sb1);
	StringBuffer sb2= new StringBuffer("Hello");
	sb2.reverse();
	System.out.println(sb2);
	StringBuffer sb3= new StringBuffer("Hello");
	sb3.deleteCharAt(3);
	System.out.println(sb3);

	}
}
```

Output -:

```
HelloJava
HJavallo
olleH
Helo
```

12)Program on Vector

```
import java.util.*;
public class Vectorexample {

	public static void main(String[] args) {
		Vector<String> vec = new Vector<String> ();
		vec.add("Tiger");
		vec.add("Cat");
		vec.add("Rat");
		vec.add("Mouse");
		System.out.println("Enter String are ==> "+vec);

		Vector<Integer> in = new Vector<Integer> ();
		in.add(1);
		in.add(2);
		in.add(3);
		in.add(4);
		in.add(5);
		System.out.println("Enter numbers are ==> "+in);
	}

}
```

Output -:

```
Enter String are ==> [Tiger, Cat, Rat, Mouse]
Enter numbers are ==> [1, 2, 3, 4, 5]
```

Another way to run a Program on Vector

```
import java.util.*;
public class Vector_Example {

	public static void main(String[] args) {
		Vector v = new Vector();
		v.add("Shekhar");
		v.add(11);
		v.add(12);
		v.add(2002);
		System.out.println("Your name & DOB is ==> "+v);
	}
}
```

Output -:

```
Your name & DOB is ==> [Shekhar, 11, 12, 2002
```

13)Single Inheritance

```
class College{
        String color;
        void roll()
        {
                System.out.println("Your roll no. is ==> 39");
        }
}
class Inherite extends College {
void cname() {
        System.out.println("College name is RCPIT");
}
        public static void main(String[] args) {
                Inherite obj =new Inherite();    //Child class Object
                obj.roll();                                //Parent class method
                obj.color="White";                 //Parent class Data
                System.out.println(obj.color);
                obj.cname();                     // Child class method
        }
}
```

Output -:

```
Your roll no. is ==> 39
  White
 College name is RCPIT
```

//Multilevel Inheritance

```
class Animal{                       // Parent class
        String color;               //Data
        void eat()                  //Method
        {
                System.out.println("Animal eat");
        }
}
class Dog extends Animal{           // Child class
        String tailshape;
        void bark()
        {
                System.out.println("Dog bark");
        }
}
class Puppy extends Dog{                    //Child class
        void play()
        {
                System.out.println("Puppy is playing");
        }
}
class Inheritance {
        public static void main(String[] args) {
                Puppy obj =new Puppy();          //Child class object
                obj.eat();                       //Parent class Method
                obj.bark();                      //Child class Method
                obj.play();
                obj.color = "Yellow";            //Parent class data
                System.out.println(obj.color);
                obj.tailshape = "Rounded";
                System.out.println(obj.tailshape);
        }
}
```

Output -:

```
 Animal
Dog bark
Puppy is playing
 Yellow
 Rounded
```

14)Program on Abstract class

```java
abstract class Bank{
        abstract int rateofintrest();
}
class HDFC extends Bank{
        int rateofintrest()
        {
                return(7);
        }
}
class SBI extends Bank{
        int rateofintrest()
        {
                return(10);
        }
}
class UBI extends Bank{
        int rateofintrest()
        {
                return(5);
        }
}
class ICICI extends Bank{
        int rateofintrest()
        {
                return(9);
        }
}
public class Abstract {
        public static void main(String[] args) {
        //Below is taking a Reference of parent class (abstract class) and object is child class
        Bank obj = new HDFC();
        System.out.println("The rate of intrest of HDFC bank is = "+obj.rateofintrest());
        Bank obj1 = new SBI();
        System.out.println("The rate of intrest of SBI bank is = "+obj1.rateofintrest());
        Bank obj3 = new UBI();
        System.out.println("The rate of intrest of UBI bank is = "+obj3.rateofintrest());
        Bank obj4 = new ICICI();
        System.out.println("The rate of intrest of ICICI bank is = "+obj4.rateofintrest());
        }
}
```

Output -:

```
The rate of intrest of HDFC bank is = 7
The rate of intrest of SBI bank is = 10
The rate of intrest of UBI bank is = 5
The rate of intrest of ICICI bank is = 9
```

15) Interface in java

```
interface Printable{
	void print();
}
class Java implements Printable
{
	public void print()
	{
		System.out.println("I am from Maharashtra");
	}
}
public class Interface implements Printable{
	public void print()
	{
		System.out.println("My Country name is India");
	}
	public static void main(String[] args) {
		Interface obj =new Interface();
		obj.print();
		Java obj1 =new Java();
		obj1.print();
	}
}
```

Output -:

```
My Country name is India
I am from Maharashtra
```

// Multiple Inheritance using interface

```
interface Eclipse{
	void print();
	}
interface Showable{
	void show();
}
class Multiple_Inheritance implements Eclipse,Showable{
	public void print()
	{
		System.out.println("I am watching a TV");
	}
	public void show()
	{
		System.out.println("I am doing a coding");
	}
	public static void main(String[] args) {
	Multiple_Inheritance obj = new Multiple_Inheritance();
	obj.print();
	obj.show();
	}
}
```

Output -:

```
I am watching a TV
I am doing a coding
```

16)Exception Handling in java

```java
import java.util.InputMismatchException;
import java.util.Scanner;
public class Exception {
    public static void main(String[] args) {
        while(true)
        {
        try
        {
            System.out.println("Enter a value of A ==> ");
            Scanner sc =new Scanner(System.in);
            int n = sc.nextInt(); //raises exception
            int result = 50/n;    // Raises Exception
            System.out.println("The result is = "+result);
            break;
        }    //try ends here
        catch(ArithmeticException ae)
        {
            System.out.println("You can't divide by zero");
        }
        catch(InputMismatchException ime)
        {
            System.out.println("Only integer value is allowed");
        }
        }    // While ends here
    }
}
```

Output -:

```
 Enter a value of A ==>
5.5
Only integer value is allowed
Enter a value of A ==>
0
You can't divide by zero
Enter a value of A ==>
5
The result is = 10
```

// Array Index Exception

```
public class Array_Exception {
     public static void main(String[] args) {
     try
     {
          int a[]= {1,2,3,4,5};
          System.out.println(a[5]);  //raises exception
     }
     catch(ArrayIndexOutOfBoundsException aie)
     {
          System.out.println("Index of array size is exceed limit");
     }



finally  // Finally will be executed finally
     {
          String M = " Maharashtra ";
          System.out.println(M);
     }
     }
}
```

Output -: Index of array size is exceed limit

// Null pointer exception

```
public class Throw {
     public static void main(String[] args) {
          try
          {
          String s = null;
          System.out.println(s.length());
          }
          catch(NullPointerException npe)
          {
               System.out.println("Can not point null string");
          }
     }
}
```

Output -: Can not point null string