```
import java.io.*;
class swap
{
public static void main(String args[])
{
int a=10,b=20,c;
System.out.println("before swapping a is :

"+a);
System.out.println("before swapping b is :

"+b);

c=a;
a=b;
b=c;

System.out.println("after swapping a is : "+a);
System.out.println("after swapping b is : "+b);

}
}
```

```
import java.io.*;
```

```
import java.util.*;
class swap
{
public static void main(String args[])
{
Scanner s=new Scanner(System.in);
int a,b,c;
System.out.println("Enter the value of A ");
a=s.nextInt();
System.out.println("Enter the value of B ");
b=s.nextInt();
System.out.println("before swapping a is : "+a);
System.out.println("before swapping b is : "+b);
c=a;
a=b;
b=c;

System.out.println("after swapping a is : "+a);
System.out.println("after swapping b is : "+b);

}
}



import java.io.*;
```

```
import java.util.*;

class swap

{

public static void main(String args[])

{

Scanner s=new Scanner(System.in);

int a,b;

System.out.println("Enter the value of A ");

a=s.nextInt();

System.out.println("Enter the value of B ");

b=s.nextInt();

System.out.println("before swapping a is : "+a);

System.out.println("before swapping b is : "+b);


a=a+b;

b=a-b;

a=a-b;


System.out.println("after swapping a is : "+a);

System.out.println("after swapping b is : "+b);


}

}
```

```
import java.util.*;
public class test
{
public static void main(String[] args)
{
Scanner s=new Scanner(System.in);
int number, reverse = 0;
System.out.println("Enter the value to reverse ");
number=s.nextInt();

while(number != 0)
{
int remainder = number % 10;
reverse = reverse * 10 + remainder;
number = number/10;
}
System.out.println("The reverse of the given number is: " + reverse);
}
}
public class test
{
public static void main(String[] args)
{
int number = 987654, reverse = 0;
while(number != 0)
```

```
{
int remainder = number % 10;
reverse = reverse * 10 + remainder;
number = number/10;
}
System.out.println("The reverse of the given number is: " + reverse);
}
}

class test{
public static void main(String args[])
{
 int n1=0,n2=1,n3,i,count=100;
 System.out.print(n1+" "+n2);//printing 0 and 1

 for(i=2;i<count;++i)//loop starts from 2 because 0 and 1 are already
printed
 {
  n3=n1+n2;
  System.out.print(" "+n3);
  n1=n2;
  n2=n3;
 }

}}
```

```
public class test
{
 public static void main(String args[])
{
  int i,m=0,flag=0;
  int n=5;//it is the number to be checked
  m=n/2;
  if(n==0||n==1){
   System.out.println(n+" is not prime number");
  }
else
{
   for(i=2;i<=m;i++){
    if(n%i==0){
     System.out.println(n+" is not prime number");
     flag=1;
     break;
    }
   }
   if(flag==0)  { System.out.println(n+" is prime number"); }
  }//end of else
}
```

```
}



import java.util.*;
public class test
{
 public static void main(String args[])
{

Scanner s=new Scanner(System.in);
  int i,m=0,flag=0;
System.out.println("Enter the no to find prime or not ");
  int n=s.nextInt();;
  m=n/2;
  if(n==0||n==1)
{
   System.out.println(n+" is not prime number");
  }
else
{
   for(i=2;i<=m;i++){
    if(n%i==0){
     System.out.println(n+" is not prime number");
     flag=1;
     break;
```

```
    }
   }
   if(flag==0)
{
System.out.println(n+" is prime number"); }
  }//end of else
}
}
```

```
class test
{
public static void main(String[] args)
{
int i,j;
for(i=1; i<=6; i++)
{
for(j=1; j<i; j++)
{
System.out.print("*");
}
System.out.println();
}
}
```

```
}
// for more patterns   https://www.javatpoint.com/how-to-print-pattern-in-java


public class test
{
   public static void main(String[] args)
{
       double radius = 7.5;
       double perimeter = 2 * Math.PI * radius;
       double area = Math.PI * radius * radius;


       System.out.println("Perimeter is = " + perimeter);
       System.out.println("Area is = " + area);
   }
}


import java.util.*;
public class test
{
   public static void main(String[] args)
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter the Radius = " );
       double radius =sc.nextDouble();
       double perimeter = 2 * Math.PI * radius;
```

```
        double area = Math.PI * radius * radius;


        System.out.println("Perimeter is = " + perimeter);
        System.out.println("Area is = " + area);
    }
}


public class test
{
    public static void main(String args[])
    {
    int width=5;
    int height=10;
    int area=width*height;
        System.out.println("Area of rectangle="+area);
     }
}




import java.util.Scanner;


class test
{
   public static void main(String args[])
   {
```

```
    int m, n, p, q, sum = 0, c, d, k;

    Scanner in = new Scanner(System.in);
    System.out.println("Enter the number of rows and columns of first matrix");
    m = in.nextInt();
    n = in.nextInt();

    int first[][] = new int[m][n];

    System.out.println("Enter the elements of first matrix");

    for ( c = 0 ; c < m ; c++ )
      for ( d = 0 ; d < n ; d++ )
        first[c][d] = in.nextInt();

    System.out.println("Enter the number of rows and columns of second matrix");
    p = in.nextInt();
    q = in.nextInt();

    if ( n != p )
      System.out.println("Matrices with entered orders can't be multiplied with
each other.");
    else
    {
      int second[][] = new int[p][q];
```

```
int multiply[][] = new int[m][q];

System.out.println("Enter the elements of second matrix");

for ( c = 0 ; c < p ; c++ )
  for ( d = 0 ; d < q ; d++ )
    second[c][d] = in.nextInt();

for ( c = 0 ; c < m ; c++ )
{
  for ( d = 0 ; d < q ; d++ )
  {
    for ( k = 0 ; k < p ; k++ )
    {
      sum = sum + first[c][k]*second[k][d];
    }

    multiply[c][d] = sum;
    sum = 0;
  }
}

System.out.println("Product of entered matrices:-");

for ( c = 0 ; c < m ; c++ )
```

```
        {
          for ( d = 0 ; d < q ; d++ )
            System.out.print(multiply[c][d]+"\t");

          System.out.print("\n");
        }
      }
    }
}


class test1
{
        static int num1=10;
        static int num2=5;
}

class test extends test1
{
        public static void main(String[] args)
{
        int num3=2;
        int result=num1+num2+num3;
        System.out.println("Result of child class is "+result);
}
```

```
}



interface Car
{
   int  speed=60;
   public void distanceTravelled();
}
interface Bus
{
   int distance=100;
   public void speed();
}
public class test  implements Car,Bus
{
   int distanceTravelled;
   int averageSpeed;
   public void distanceTravelled()
   {
      distanceTravelled=speed*distance;
      System.out.println("Total Distance Travelled is : "+distanceTravelled);
   }
   public void speed()
   {
      int  averageSpeed=distanceTravelled/speed;
```

```
        System.out.println("Average Speed maintained is : "+averageSpeed);
    }
    public static void main(String args[])
    {
        test t1=new test();
        t1.distanceTravelled();
        t1.speed();
    }
}



class test
{
 test()
{
 System.out.println("Hello");
}
public static void main(String args[])
{
 test t1=new test();
 t1=null;
 System.gc();
}
 public void finalize()
{
```

```
 System.out.println("Destroyed");
}
}

class Human
{
  //Overridden method
  public void eat()
  {
    System.out.println("Human is eating");
  }
}
class Boy extends Human{
  //Overriding method
  public void eat()
{
    System.out.println("Boy is eating");
  }
  public static void main( String args[])
{
    Boy obj = new Boy();
    //This will call the child class version of eat()
    obj.eat();
  }
}
```

```
class test
{
 public static void main(String args[])
 {
  int a,b,c;
  try
  {
   a=0;
   b=10;
   c=b/a;
   System.out.println("This line will not be executed");
  }
  catch(ArithmeticException e)
  {
   System.out.println("Divided by zero");
  }
  System.out.println("After exception is handled");
 }
}

public class test extends Thread {

  public void run()
{
```

```
        System.out.println("Thread is runing !!");
    }

    public static void main(String[] args)
{
        test t1 = new test();
        test t2 = new test();

        System.out.println("T1 ==> " + t1.getState());
        System.out.println("T2 ==> " + t2.getState());

        t1.start();
        System.out.println("T1 ==> " + t1.getState());
        System.out.println("T2 ==> " + t2.getState());

        t2.start();
        System.out.println("T1 ==> " + t1.getState());
        System.out.println("T2 ==> " + t2.getState());
    }
}
```