// 1. program to demonstrate arithmetic operator

```
public class prog1 {
   public static void main(String args[])
   {int a,b,c;
      a=19;b=4;
   System.out.println("The basic java program");
      c=a+b;
      System.out.println("the sum of "+a+" and "+b +" is:"+c);
      c=a-b;
      System.out.println("the diff of "+a+" and "+b +" is:"+c);
      c=a*b;
      System.out.println("the product of "+a+" and "+b +" is:"+c);
      c=a/b;
      System.out.println("the division of "+a+" and "+b +" is:"+c);
      c=a%b;
      System.out.println("the mod of "+a+" and "+b +" is:"+c);

   }
}
```

Output

```
The basic java program
the sum of 19 and 4 is:23
the diff of 19 and 4 is:15
the product of 19 and 4 is:76
the division of 19 and 4 is:4
the mod of 19 and 4 is:3

Process finished with exit code 0
```

//2.program to demonstrate various operators

```
public class prog2 {
   public static void main(String args[])
   {int a,b,c;
      a=19;b=4;
      System.out.println("Arithmetic operator:");
      System.out.println("The basic java program");
      c=a+b;
      System.out.println("the sum of "+a+" and "+b +" is:"+c);
      c=a-b;
      System.out.println("the diff of "+a+" and "+b +" is:"+c);
      c=a*b;
      System.out.println("the product of "+a+" and "+b +" is:"+c);
      c=a/b;
      System.out.println("the division of "+a+" and "+b +" is:"+c);
      c=a%b;
      System.out.println("the mod of "+a+" and "+b +" is:"+c);

      int i, j;
      boolean b1, b2;
      i = 10;j = 11;
      System.out.println("\nRelational operator:");
      if(i < j) System.out.println("i < j");
      if(i <= j) System.out.println("i <= j");
      if(i != j) System.out.println("i != j");
      if(i == j) System.out.println("this won't execute");
      if(i >= j) System.out.println("this won't execute");
      if(i > j) System.out.println("this won't execute");
      b1 = true;b2 = false;

      System.out.println("\nLogical operator:");
      if(b1 & b2) System.out.println("this won't execute");
      if(!(b1 & b2)) System.out.println("!(b1 & b2) is true");
      if(b1 | b2) System.out.println("b1 | b2 is true");
      if(b1 ^ b2) System.out.println("b1 ^ b2 is true");
      if(i != 0 && (i % j) == 0) //short circuit and
         System.out.println(j + " is a factor of " + i);

      System.out.println("\nBitwise operator:");
      byte val1,val2;
      val1=101; val2=100;
      System.out.println("Bitwise and "+val1+" &"+val2+" :"+(val1&val2));
      System.out.println("Bitwise or "+val1+" |"+val2+" :"+ (val1|val2));
      System.out.println("Bitwise xor "+val1+" ^"+val2+" :"+ (val1^val2));
      System.out.println("Bitwise not "+val1+" :"+ (~val1));
      System.out.println("Bitwise right shift "+val1+" :"+ (val1>>1));
      System.out.println("Bitwise left shift "+val1+" :"+ (val1<<1));
         }
}
```

Output

```
Arithmetic operator:
The basic java program
the sum of 19 and 4 is:23
the diff of 19 and 4 is:15
the product of 19 and 4 is:76
the division of 19 and 4 is:4
the mod of 19 and 4 is:3

Relational operator:
i < j
i <= j
i != j

Logical operator:
!(b1 & b2) is true
b1 | b2 is true
b1 ^ b2 is true

Bitwise operator:
Bitwise and 101 &100 :100
Bitwise or 101 |100 :101
Bitwise xor 101 ^100 :1
Bitwise not 101 :-102
Bitwise right shift 101 :50
Bitwise left shift 101 :202
```

//3.Program to demonstrate various loop

```
public class prog3 {
   public static void main(String args[])
   {int i=0;
      for (i=0;i<10;i++)
      System.out.println("Value  inside for loop:"+i);
      System.out.println("\nValue outside for loop:"+i+"\n");
      while (i>5) {
         System.out.println("Value inside while loop:"+i);
      i--;}
      System.out.println("\nValue outside while loop:"+i+"\n");
      do {
         System.out.println("Value inside do while loop:" + i);
         i--;
      }while (i>0);
      System.out.println("\nValue outside do while loop:"+i+"\n");
   }
}
```

Output

```
Value  inside for loop:0
Value  inside for loop:1
Value  inside for loop:2
Value  inside for loop:3
Value  inside for loop:4
Value  inside for loop:5
Value  inside for loop:6
Value  inside for loop:7
Value  inside for loop:8
Value  inside for loop:9

Value outside for loop:10

Value inside while loop:10
Value inside while loop:9
Value inside while loop:8
Value inside while loop:7
Value inside while loop:6

Value outside while loop:5

Value inside do while loop:5
Value inside do while loop:4
Value inside do while loop:3
Value inside do while loop:2
Value inside do while loop:1

Value outside do while loop:0
```

//4.program to demonstrate array in java

```
public class prog4 {
   public static void main(String[] args) { // different ways to initialize

      //1d array
      int[] arr1d = new int[10];
      arr1d = new int[]{1, 2, 3, 4, 12, 34, 3, 2, 2, 3};
      int[] arr1d2 = {1, 2, 2, 2, 1, 1, 1, 3, 3};
      int x, i = 0;
      //traversal via simple for loop
      for (i = 0; i < 10; i++) {
         x = arr1d[i];
         System.out.println("Value of element arr1d[" + i + "]:" + x);
      }
      for (i = 0; i < arr1d2.length; i++) {
         x = arr1d2[i];
         System.out.println("Value of element arr1d2[" + i + "]:" + x);
      }

      // 2d array
      int[][] arr2d = {{1, 2, 3}, {4, 5, 6}}; //a common array
      int[][] arr2dirr = new int[3][];  // irregular 2 d array
      arr2dirr[0] = new int[]{1, 2, 3, 4};
      arr2dirr[1] = new int[]{1, 3, 4};
      arr2dirr[2] = new int[]{1, 4};
      System.out.println("Printing 2 d array \n");
      for (int[] j : arr2d) {
         System.out.println();
         for (int y : j) {
            System.out.print(y + " ");
         }
      }
      System.out.println("\nPrinting another 2 d array \n");
      for (int[] j : arr2dirr){
         System.out.println();
         for (i=0;i< j.length;i++)
         { x=j[i];
            System.out.print(x+" ");
         }
      }
   }

}
```

Output

```
Value of element arr1d[0]:1
Value of element arr1d[1]:2
Value of element arr1d[2]:3
Value of element arr1d[3]:4
Value of element arr1d[4]:12
Value of element arr1d[5]:34
Value of element arr1d[6]:3
Value of element arr1d[7]:2
Value of element arr1d[8]:2
Value of element arr1d[9]:3
Value of element arr1d2[0]:1
Value of element arr1d2[1]:2
Value of element arr1d2[2]:2
Value of element arr1d2[3]:2
Value of element arr1d2[4]:1
Value of element arr1d2[5]:1
Value of element arr1d2[6]:1
Value of element arr1d2[7]:3
Value of element arr1d2[8]:3
Printing 2 d array


1 2 3
4 5 6
Printing another 2 d array


1 2 3 4
1 3 4
1 4
```

//5.program to demonstrate strings

```
public class prog5 {
   public static void main(String[] args)
   { String str=new String("this is str");
      String str2="this is str2";
      String str3=str2;
      String str4=args[args.length-1];// reads last string of cmd line arguments
      System.out.println(str4);
      System.out.println("the string are: "+str+"  "+str2+"  "+str3);
      System.out.println("boolean str.equals(str2)"+ str.equals(str2));
      System.out.println("str.length():"+str.length());
      System.out.println("str.charAt(2)"+str.charAt(2));
      System.out.println("str.compareTo(str3)"+str.compareTo(str3));
      System.out.println("str.indexOf(\"is\")"+str.indexOf("is"));
      System.out.println("str.lastindexOf(\"is\")"+str.lastIndexOf("is"));
   }
}
```

Output

```
F:\codes\java codes\java_codes_coll\out\production\java_codes_coll>java prog5 abcd
abcd
the string are: this is str  this is str2  this is str2
boolean str.equals(str2)false
str.length():11
str.charAt(2)i
str.compareTo(str3)-1
str.indexOf("is")2
str.lastindexOf("is")5
```

//6.program to demonstrate class

```
class cars
{   int milage,price;
    String name;
     cars() // constructor
     {milage=0;
     price=0;
     name="NA";
        System.out.println("simple constructor invoked");}

    cars(int i,int j,String str) //parameterized constructor
    {milage=i;
    price=j;
    name=str;
       System.out.println("parametrized constructor invoked");}

    cars(cars a)
    {milage=a.milage;
    price=a.price;
    name=a.name;
       System.out.println("Copy constructor invoked");}

    void display()
    {System.out.println("the details are:");
       System.out.println(name+" "+ milage+" "+ price);
    }

    int range(int i){return milage*i;}

}
public class prog6 {
    public static void main(String[] args) {
       cars a = new cars();
       a.display();
       cars b = new cars(18, 240000, "alto");
       b.display();
       cars c = new cars(b);
       c.display();
       System.out.println("range of c in 5ltr fuel is"+c.range(5));
    }


}
```

```
simple constructor invoked
the details are:
NA 0 0
parametrized constructor invoked
the details are:
alto 18 240000
Copy constructor invoked
the details are:
alto 18 240000
range of c in 5ltr fuel is90
```

//7.program to demonstrate inheritance

```
class cars
{       int milage,price;
        String name;
        cars() // constructor
        {milage=0; price=0;  name="NA";
        System.out.println("cars simple constructor invoked");}

        cars(int i,int j,String str) //parameterized constructor
        {milage=i;   price=j;  name=str;
        System.out.println("cars parametrized constructor invoked");}

        cars(cars a)
        {milage=a.milage; price=a.price;   name=a.name;
        System.out.println("cars Copy constructor invoked");}

        void display()
        {System.out.println("the details are:\n"+name+" "+ milage+" "+ price); }

        int range(int i){return milage*i;}
}
class disel extends cars
{       int power;
        static int tax;
        disel()
        {super(18,500000,"nexon");
        power=100;  tax=5;
        System.out.println("Constructor of disel class invoked");
        }
        void display(){
         System.out.println("The details of disel cars milage"+milage+" price:"+ (price*(100+tax)/100));
         System.out.println(" name :"+name+" power:"+power);
        }

}
public class prog7 {
        static public void main(String [] args)
        { disel d1=new disel();
        d1.display();
        disel.tax=20;
        d1.display();}
}
```

Output

```
cars parametrized constructor invoked
Constructor of disel class invoked
The details of disel cars milage18 price:525000
 name :nexon power:100
The details of disel cars milage18 price:600000
 name :nexon power:100
```

//8.program to demonstrate method overriding and polymorphism

```
class shape{
   void show(){System.out.println("A normal shape");}
}
class twod extends shape{
   void show(){System.out.println("A 2d shape");}
}
class square extends twod{
   void show(){System.out.println("a square");}
}
public class prog8 {
   public static void main(String[] args){
      shape s=new shape();
      twod s1=new twod();
      square s2=new square();
      s.show();
      s1.show();
      s2.show();
      s=s1;
      s.show();
      s=s2;
      s.show();
   }
}
```

Output

```
A normal shape
A 2d shape
a square
A 2d shape
a square

Process finished with exit code 0
```

//9.program to demonstrate package in java

```
import mypack.books;
public class prog9 {
   public static void main(String [] args)
   {books b=new books();
   b.show();}
}
```

```
package mypack;

public class books {
   int price;
   String name;
  public books(){name="java"; price=100;}
   public void show(){System.out.println(name+" costs"+price);}
}
```

Output

```
java costs100

Process finished with exit code 0
```

//10.program to demonstrate interface

//area.java

```
public interface area {
   double pi = 3.14;

   void totalarea(int i);
}
```

```
//prog10.java
class circle implements area {
   circle(){totalarea(5);}
   public void totalarea(int r)
   {System.out.println("the total area is:"+pi*r*r);}
}
public class prog10{
  public static void main(String [] args)
  {circle c=new circle();}
}
```

Output

```
the total area is:78.5

Process finished with exit code 0
```

//11.program to demonstrate inherited interface

```
//prog11.java
class account implements loan{
   int balance;

   public void getbalance() {
      System.out.println("account opened with 1000 rupees");
      balance=1000;}

   public int showbalance(int j) {
      balance=j;
      return j-250;}

   public int totalloan(int p, int t) {
      int amount=p+p*rate*t/100;
      if (balance*5<amount)
      {System.out.println("sorry loan cant be approved");
         return 0;}
      else
      {System.out.println("loan approved with amount of"+amount);
         return amount;}
   }
   public int si(int p,int t)
   {int si=totalloan(p,t)-p;
      return si;
   }

}
public class prog11 {
   public static void main(String[] args) {
      account a = new account();
      a.getbalance();
     System.out.println("available balance is"+ a.showbalance(2000));
     System.out.println("Loan status with si of :"+ a.si(3000,3));
   }
}

//bank.java
public interface bank {
int rate=10;
public void getbalance();
int showbalance(int j);
}

//loan.java

public interface loan extends bank {
  public int totalloan(int p, int t);
   int si(int p, int t);
}
```

Output

```
account opened with 1000 rupees
available balance is1750
loan approved with amount of3900
Loan status with si of :900

Process finished with exit code 0
```

//12.program to demonstrate exception handling in java

```
public class prog12 {
   public static void main(String[] args) {
      int[] arr = new int[12];
      // a simple exception block
      try {
         System.out.println("inside try block");
         arr[13] = 10;
      } catch (ArrayIndexOutOfBoundsException exc) {
         System.out.println("exception caught" + exc);
      }


      arr = new int[]{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
      //multicatch try block with final
      try {
         for (int i = 11; i > 0; i--) {
            arr[i] = arr[i] / (i-2);
            System.out.println("at position " + i + "new value " + arr[i]);
         }
      } catch (ArithmeticException | ArrayIndexOutOfBoundsException exc) {
         System.out.println(exc);
      }
   }
}
```

Output

```
inside try block
exception caughtjava.lang.ArrayIndexOutOfBoundsException: Index 13 out of bounds for length 12
at position 11new value 1
at position 10new value 1
at position 9new value 1
at position 8new value 1
at position 7new value 1
at position 6new value 1
at position 5new value 2
at position 4new value 2
at position 3new value 4
java.lang.ArithmeticException: / by zero

Process finished with exit code 0
```

//13.program to demonstrate exception throw \throws and user defined exception

```
import java.io.IOException;
class myexc extends Exception
{ public String toString()
   {return " error its an odd no";}
}


public class prog13 {
   static void mynum() throws myexc, IOException
   {int i=System.in.read();
      if (i%3!=0)
         throw new myexc();}


   public static void main(String[] args)
   { int i,j;
      i=2; j=3;
      // a normal throw
      try
      {if (i%3==0|j%3==0)
         throw new myexc();}
      catch (myexc exc)
      {System.out.println(exc);}

      //catching exception thrown by function
      try{mynum();}
      catch(Exception exc)
      {System.out.println(exc);}

}

}
```

Output

```
 error its an odd no
123
 error its an odd no

Process finished with exit code 0
```

//14.program to demonstrate read and write on console

```
import java.io.IOException;

public class prog14 {
   public static void main (String[] args) throws IOException
   {byte[] b=new byte[10];
      System.out.println("write something");
      System.in.read(b);
      for (int i=0;i<b.length;i++)
         System.out.print((char)b[i]);
   }
}
```

Output

```
"F:\User programs\java\bin\java.exe" "-javaagent:F:\User programs\JetBrains\IntelliJ IDEA Common
write something

1223443

1223443
0
Process finished with exit code 0
```

//15. program to demonstrate file io using byte and character stream

```
import java.io.*;

class byteio {
   void iowrite()
   { try {var fout = new FileOutputStream("bfile", true);
         int b = 100;
         System.out.println("Writing in bfile");
         fout.write(b);
         fout.close();
      } catch (FileNotFoundException exc) { System.out.print(exc); }
            catch (IOException exc) {System.out.print("write error" + exc);}
   }

   void ioread()
 {  try {
         var fin = new FileInputStream("bfile");
         int b = fin.read();
         System.out.println("Value read from bfile is :"+b);
         fin.close();
      } catch (IOException exc) { System.out.print("read error " + exc);}
   }

   void Binaryread() {
      int i = 0;
      double d = 0;
      try {
         var fin = new DataInputStream(new FileInputStream("Bifile"));
         i = fin.readInt();
         d = fin.readDouble();
         System.out.println("Values read from Bfile are" + i + " " + d);
         fin.close();
      } catch (IOException exc) {  System.out.println("ERROR in Write " + exc);}
   }

   void Binarywrite() {
      int i = 100;
      double d = 100.00;
      try {
         var fout = new DataOutputStream(new FileOutputStream("Bifile", true));
        System.out.println("writing in Bfile");
         fout.writeInt(i);
         fout.writeDouble(d);
         fout.close();
      } catch (IOException exc) {System.out.println("ERROR in Write " + exc);}
   }
}

class Chario
{ void charwrite() throws IOException {
   BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
```

```
   int i;
   String str;
   System.out.println("Please enter a int and string");
   //i = Integer.parseInt(br.readLine());
   str = br.readLine();
   i= Integer.parseInt(str);
   str = br.readLine();
   System.out.println("writing to charfile "+i+"  "+str);
   try {
      FileWriter fin = new FileWriter("charfile");
      fin.write(Integer.toString(i));
      fin.write("\n"+str);
      System.out.println("Writing in charfile");
      fin.close();
   } catch (IOException exc) { System.out.println("Something happened while writing " + exc);}
   }

   void charread()
   {String str;
      int i=0;
      try {
         BufferedReader br = new BufferedReader(new FileReader("charfile"));
        System.out.println("Reading from charfile");
        i= Integer.parseInt(br.readLine());
         System.out.println(i);
         str= br.readLine();
         System.out.println(str);
      }catch (IOException exc){System.out.println("ERROR while reading"+exc);}
   }
}



public class prog15 {
   public static void main(String[] arg)
   {   byteio obj=new byteio();
      obj.iowrite();
      obj.ioread();
      obj.Binarywrite();
      obj.Binaryread();

      Chario cobj=new Chario();
      try{cobj.charwrite();}
      catch (IOException exc){System.out.println("error in write");}
      cobj.charread();
   }
}
```

Output

```
Writing in bfile
Value read from bfile is :100
writing in Bfile
Values read from Bfile are100 100.0
Please enter a int and string
123
abcd
writing to charfile 123  abcd
Writing in charfile
Reading from charfile
123
abcd

Process finished with exit code 0
```

//16.program to demonstrate creation of thread

```
class mythread implements Runnable {
   String thrdname;
   mythread(String str) {
      thrdname = str;
   }

   public void run() {
      System.out.println("running thread " + thrdname);
      try {
         for (int i = 0; i < 10; i++) {
            Thread.sleep(1000);
            System.out.println("running thread " + thrdname + " time " + i);
         }
      } catch (InterruptedException exc) {
         System.out.println("unable to halt thread" + exc);
      }
   }
}

public class prog16 {
public static void main(String [] args)
{mythread mt=new mythread("thrd1");
Thread newthrd=new Thread(mt);
   mythread mt2=new mythread("thrd2");
   Thread newthrd2=new Thread(mt2);
newthrd.start();
newthrd2.start();}
   }
```

Output

```
running thread thrd1
running thread thrd2
running thread thrd1 time 0
running thread thrd2 time 0
running thread thrd1 time 1
running thread thrd2 time 1
running thread thrd1 time 2
running thread thrd2 time 2
running thread thrd1 time 3
running thread thrd2 time 3
running thread thrd1 time 4
running thread thrd2 time 4
running thread thrd1 time 5
running thread thrd2 time 5
running thread thrd1 time 6
running thread thrd2 time 6
running thread thrd1 time 7
running thread thrd2 time 7
running thread thrd1 time 8
running thread thrd2 time 8
running thread thrd1 time 9
running thread thrd2 time 9

Process finished with exit code 0
```

//17.program to demonstrate thread priority and some thread class function

```
//creating thread via Runnable
class mythd implements Runnable
{String name;
   Thread thr;

   mythd(String str)
   {name=str;
      thr=new Thread(this,name);}

   public static mythd createandrun(String str)
   {mythd m1=new mythd(str);
      m1.thr.start();
      return m1;
   }

   public void run() {

      try {
         for (int i = 0; i < 10; i++) {
            Thread.sleep(1000);
            System.out.println("running thread " + thr.getName() + " time " + i);
            System.out.println("running " + thr.getName() + " with priority " + thr.getPriority());
         }
      } catch (InterruptedException exc) {
         System.out.println("unable to halt thread" + exc);
      }
   }
}

//creating thread via Thread
class myth extends Thread
{   myth(String str)
   {super(str);}

   public void run()
   {try {
      for (int i = 0; i < 10; i++) {
         Thread.sleep(1000);
         this.setPriority(i+1);
         System.out.println("running "+this.getName()+" with priority "+this.getPriority()+"  via Thread
class");
         System.out.println("running thread " + this.getName() + " time " + i);
      }
   } catch (InterruptedException exc)
```

```
        {System.out.println("unable to halt thread" + exc);}

    }

}
public class prog17 {
    public static void main(String[] args) throws InterruptedException {
        mythd mt = mythd.createandrun("child1");
        myth mt2 = new myth("child2");
        System.out.println("child 1 status "+mt.thr.isAlive());
        mt2.start();
        mt2.join();
        System.out.println("mt2 is dead");
        System.out.println("child 1 status "+mt.thr.isAlive());
    }
}
```

Output

```
running thread child2 time 3
running thread child1 time 4
running child1 with priority 5
running child2 with priority 5  via Thread class
running thread child2 time 4
running thread child1 time 5
running child1 with priority 5
running child2 with priority 6  via Thread class
running thread child2 time 5
running thread child1 time 6
running child1 with priority 5
running child2 with priority 7  via Thread class
running thread child2 time 6
running thread child1 time 7
running child1 with priority 5
running child2 with priority 8  via Thread class
running thread child2 time 7
running thread child1 time 8
running child1 with priority 5
running child2 with priority 9  via Thread class
running thread child2 time 8
running thread child1 time 9
running child1 with priority 5
running child2 with priority 10  via Thread class
running thread child2 time 9
mt2 is dead
child 1 status false

Process finished with exit code 0
```

//18.program to demonstrate synchronized method

```
class pattern {
   synchronized static void display(int i)
   { System.out.println(Thread.currentThread().getName());
      for(int j=0;j<5;j++){
         System.out.print(i*j+" ");
         try {Thread.sleep(500);
         } catch (InterruptedException e) {System.out.println(e);}
      }
   }
}

class mt implements Runnable
{Thread t;
   int i;
   mt(String str,int i)
   {this.i=i;
      t=new Thread(this,str);}

   public static mt candr  (String str,int i)
   {mt c1=new mt(str,i);
      c1.t.start();
      return c1;
   }

   public void run()
   {System.out.println("starting "+Thread.currentThread().getName());
      pattern.display(i);
      System.out.println(t.getName() + " terminating.");
   }
}

public class prog18 {
   public static void main(String[] args)
   {mt t1=mt.candr("child1",1);
      mt t2=mt.candr("child2",3);
   }
}
```

Output

```
starting child1
starting child2
child1
0 1 2 3 4 child2
0 child1 terminating.
3 6 9 12 child2 terminating.

Process finished with exit code 0
```