java lab

factorial

```
import java.util.*;
class factor{
    public static void main(String args[]){
        System.out.println(" give number");
        Scanner s=new Scanner(System.in);
        int n=s.nextInt();
        int ans=1;
        while(n>0){
            ans*=n;
            n--;
        }
        System.out.println("factorial will be "+ans);
    }
}
```

operators

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

        System.out.println(" give two number");
        Scanner s=new Scanner(System.in);
        int a=s.nextInt();
        int b=s.nextInt();
        System.out.println("a is "+a);
        System.out.println("b is "+b);
        System.out.println(" a+b="+(a+b));
        System.out.println(" a-b="+(a-b));
        System.out.println(" a*b="+(a*b));
        System.out.println(" a/b="+(a/b));
        System.out.println(" a%b="+(a%b));
        System.out.println(" a>b="+(a>b));
        System.out.println(" a=b="+(a==b));
        System.out.println(" a!=b="+(a!=b));
        System.out.println(" a&b="+(a&b));
        System.out.println(" a|b="+(a|b));

    }
}
```

constructor

```
class rect{
    int l,b;
    rect(int x,int y){
```

```
        l=x;
        b=y;
    }
    int area(){
        return l*b;
    }
}
class construct{
    public static void main(String args[]){
        rect r1=new rect(10,20);
        int a=r1.area();
        System.out.println(" area of recangle ="+a);
    }
}
```

overloading

```
//overloading
import java.util.*;
class test{
     public int area(int a,int b){
        return a*b;
    }
    public int area(int a){
        return a*a;
    }
}
 public class overload{

    public static void main(String args[]){
        test a1=new test();
        System.out.println(" area of rectancle of length 10 and breadth
5="+a1.area(10,5));
        System.out.println("area of square of lenth 10 ="+a1.area(10));
    }
}
```

inheritence

```
class A{
    public void f1(){
        System.out.println("A");
    }
}
class B extends A{
    public void f2(){
        System.out.println("B");
    }
}
public class inherit{
    public static void main(String args[]){
        B b1=new B();
```

```
        b1.f1();
        b1.f2();
    }
}
```

string

```
//string operations
class str{
    public static void main(String args[]){
        String s1="sumit";
        String s2="kumar";
        String s3=s1.toUpperCase();
        System.out.println("concatenation="+(s1.concat(s2)));
        System.out.println(" uppercase="+(s1.toUpperCase()));
        System.out.println(" equality="+(s1.equals(s2)));
        System.out.println(" length="+(s1.length()));
        System.out.println("char at 1 position in s1="+(s1.charAt(1)));
        System.out.println(" compare ="+(s1.compareTo(s2)));
        System.out.println("substring(0,3)="+(s1.substring(0,3)));
        System.out.println("position of s in s1="+(s1.indexOf('s')));
        System.out.println("replace S to s"+(s1.replace('s','S')));
        System.out.println("trim="+(s1.trim()));
        System.out.println("s3:"+s3);
    }
}
```

concatenation=sumitkumar
 uppercase=SUMIT
 equality=false
 length=5
char at 1 position in s1=u
 compare =8
substring(0,3)=sum
position of s in s1=0
replace S to sSumit
trim=sumit
s3:SUMIT

exception handling

```
class except{
    public static void main(String args[]){
        int a=10,b=0;
        System.out.println(" 10/0="+(a/b));
    }
}
```

Exception in thread "main" java.lang.ArithmeticException: / by zero

at except.main(except.java:4)

human

```
import java.awt.*;
import java.applet.*;
public class aplet extends Applet{
    public void paint(Graphics g){
        g.drawOval(40,40,120,150);
        g.drawOval(55,75,30,20);
        g.drawOval(100,75,30,20);
        g.fillOval(65,80,10,10);
        g.fillOval(120,80,10,10);
        g.drawOval(85,100,30,30);
        g.fillArc(60,125,80,40,180,180);
        g.drawOval(25,90,15,30);
        g.drawOval(160,90,15,30);
    }
}
```

exception

```
import java.util.Scanner;
public class Exception_handling
{
    public static void main(String[] args)
     {
        Scanner scanner = new Scanner(System.in);
        try
        {
            System.out.println("Enter a number: ");
            int num1 = scanner.nextInt();
            System.out.println("Enter another number: ");
            int num2 = scanner.nextInt();
            int result = num1 / num2;
            System.out.println("Result: " + result);
        }
        catch (ArithmeticException e)
        {
            System.out.println("Exception caught: " + e);
            System.out.println("Cannot divide by zero!");
        }
         catch (Exception e)
        {
            System.out.println("Exception caught: " + e);
        }
        finally
        {
            scanner.close();
            System.out.println("Program has finished executing.");
        }
    }
```

```
}
```

multiple inheritence

```
interface Animal{
    void sound();
}
interface Bird{
    void fly();
}
class Parrot implements Animal,Bird{
    //override
    public void sound(){
        System.out.println(" ktore ktore ");
    }
    public void fly(){
        System.out.println("tota ud");
    }
}
public class mul_interface{
    public static void main(String args[]){
        Parrot p1=new Parrot();
        p1.sound();
        p1.fly();
    }
}
```

interface

```
interface Area{
   float pi=3.14f;
    float sol(float a,float b);
}
class rect implements Area{
    public float sol(float a,float b){
        return a*b;
    }
}
class circ implements Area{
    public float sol(float a, float b){
        return (pi *a *b);
    }
}
public class interface_test{
    public static void main(String args[]){
        rect r1=new rect();
        System.out.println(" area of rectangle ="+r1.sol(10,20));
        circ c1=new circ();
        System.out.println(" area of circle"+c1.sol(10,10));
    }
}
```

multilevel inheritence

```
class A1{
    void f1(){
        System.out.println(" A1 is called");
    }
}
class B1 extends A1{
     void f2(){
        System.out.println(" b1 is called");
    }
}
class C1 extends B1{
    void f3(){
        System.out.println("c1 is called");
    }
}
public class mul_inherit{
    public static void main(String args[]){
        C1 c1=new C1();
        c1.f1();
    }
}
```

thread runnable

```
class A implements Runnable{
    public void run(){
        int i;
        for( i=0;i<5;i++){
            System.out.println("thread A= "+i);
        }
    }
}
class B implements Runnable{
    public void run(){
        int i;
        for(i=0;i<5;i++){
            System.out.println("thread B= "+i);
        }
    }
}
public class thread_run{
    public static void main (String args[]){
        Thread t1=new Thread(new A());
        Thread t2=new Thread(new B());
        t1.start();
        t2.start();
    }
}
```

thread extend thread

```
class A extends Thread{
    public void run(){
        for(int i=0;i<5;i++){
            System.out.println(" thread A="+i);
        }
    }
}
class B extends Thread{
    public void run(){
        for(int i=0;i<5;i++){
            System.out.println(" thread B="+i);
        }
    }
}
public class threading {
    public static void main(String args[]){
        A a1=new A();
        B b1=new B();
        a1.start();
        b1.start();
    }
}
```

making file

```
import java.io.*;
 public class ft1{
    public static void main (String args[]) throws IOException {
        File f1=new File("ft2.txt");
        f1.createNewFile();
        System.out.println(" file exists :"+f1.exists());
        System.out.println("length of file :"+f1.length());
    }
}
```

buffer writer

```
import java.io.*;
public class buffer_class{
     public static void main (String args[]) throws IOException {
        FileWriter fw=new FileWriter("ft2.txt",true);
        BufferedWriter bf =new BufferedWriter(fw);
        bf.write(" computer");
        bf.close();
    }
}
```

buffer reader

```
import java.io.*;
public class buffer_class{
     public static void main (String args[]) throws IOException {
```

```
        int ch;
        BufferedReader bf =new BufferedReader(new FileReader("ft2.txt"));
        while((ch=bf.read())!=-1){
            System.out.print((char)ch);
        }
        bf.close();
    }
}
```

throw catch exception handling

```
public class ex{
    public static void main(String args []){
         default throw default catch
        int a=10,b=0;
        System.out.println(" answer :"+(a/b));

        //default throw our catch
       try{
            int a=10,b=0;
            System.out.println("answer :"+(a/b));
        } catch(ArithmeticException e){
            System.out.println("exception:"+e.getMessage());
        }
        // our throw default catch
        int a=10,b=0;
        if(a>b){
            throw new ArithmeticException("kuchh bhi");
        }
        System.out.println("our throw default catch");
        // o/p-> Exception in thread "main" java.lang.ArithmeticException: kuchh bhi
        //our throw our catch
        try{
            int a=10,b=0;
            if(a>b){
            throw new ArithmeticException ("kuchh bhi");
            }
            System.out.println("b-a="+(b-a));
        } catch(ArithmeticException e){
            System.out.println(" exception :"+e.getMessage());
        }
        System.out.println("our throw our catch");
    }
//   o/p->   exception :kuchh bhi
// our throw our catch
}
```