

# JAVA

---

"Door Of Your Dreams"

**KR NETWORK**  
**CLOUD**

Leading Edge IT Training

Java is a **programming language** and a **platform**. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by *Sun Microsystems* (which is now the subsidiary of Oracle) in the year 1995. *James Gosling* is known as the father of Java. Before Java, its name was *Oak*. Since Oak was already a registered company, so James Gosling and his team changed the Oak name to Java.

1. Desktop Applications such as acrobat reader, media player, antivirus, etc.
2. Web Applications
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System
6. Smart Card
7. Robotics
8. Games, etc.

## Java Version History

Many java versions have been released till now. The current stable release of Java is Java SE 10.

1. JDK Alpha and Beta (1995)
2. JDK 1.0 (23rd Jan 1996)
3. JDK 1.1 (19th Feb 1997)
4. J2SE 1.2 (8th Dec 1998)
5. J2SE 1.3 (8th May 2000)
6. J2SE 1.4 (6th Feb 2002)
7. J2SE 5.0 (30th Sep 2004)
8. Java SE 6 (11th Dec 2006)
9. Java SE 7 (28th July 2011)
10. Java SE 8 (18th Mar 2014)
11. Java SE 9 (21st Sep 2017)
12. Java SE 10 (20th Mar 2018)

## Platform Independent

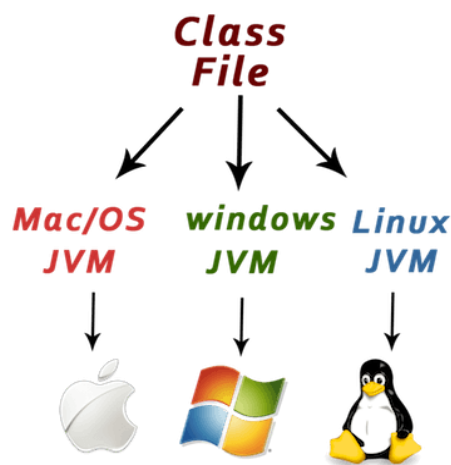
Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs.

There are two types of platforms software-based and hardware-based. Java provides a software-based platform.

The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on the top of other hardware-based platforms. It has two components:

1. Runtime Environment
2. API(Application Programming Interface)

Java code can be run on multiple platforms, for example, Windows, Linux, Sun Solaris, Mac/OS, etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms, i.e., Write Once and Run Anywhere(WORA).



## Object-oriented

Java is an object-oriented programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behavior.

Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.

Basic concepts of OOPs are:

1. Object
2. Class
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation

## First Java Program

Class atul

```
{
    public static void main(String s[])
    {
        System.out.print("Hello");
    }
}
```

Install the JDK if you don't have installed it

Set path of the jdk/bin directory.

Create the java program.

Save it as :- atul.java

Compile :- javac atul.java

Run:- java atul (run it with class file name).

## Identifiers

Identifiers are user defined names given to variables name ,function name etc. in order to refer in program.

- The first character of an identifiers should start with an alphabet or underscore.
- The identifiers should contain alphabets, digits and underscore.
- There should not be any whitespace within the identifier.
- Keywords should not be used.

Ex:-

Valid	Invalid
a	1a
Atul	1atul
A1	Atul
Atul1	
Atul_kumar	

## Keywords

Reserved word of any programming language is called Keywords.

Ex:- int, for,while,char,if etc.

## Comments

Comments are not an executable part but a comment just describes the program or provides information about each statements in the program.

Single Line Comment

```
// Comment
```

Multiline Comment

```
/** Comment */
```

## Data Type

What type of data you want to store in the program.

There are three type of data type we can use.

- 1)Primary data type
- 2)Derived data type
- 3)User-defined data type

### **1)Primary data type:-**

The data type that represent numbers and characters. Integer, float and character.

### **2)Derived data type:-**

The data type that are obtained from the primary data type such as array and structures.

### **3)User-defined data type:-**

Data type that are defined by the user like function.

Data Type	Default Value	Default size
boolean	false	1 bit
char	'\u0000'	2 byte
byte	0	1 byte

short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte

### Typecasting

class Simple

```
{
    public static void main(String[] s)
    {
        float f=10.5f;
        //int a=f;//Compile time error
        int a=(int)f;
        System.out.println(f);
        System.out.println(a);
    }
}
```

**P1) write a program to print your name.**

```
class atul
{
    public static void main(String s[])
    {
        System.out.print("Atul Kumar Varshney");
    }
}
```

**P2) write a program to add two numbers..**

```
class atul
```

```
{  
    public static void main(String s[])  
    {  
        int a=10,b=20,c;  
        c=a+b;  
        System.out.print ("Sum"+c);  
    }  
}
```

**P3) Write a program to multiply two values.**

```
class atul  
{  
    public static void main(String s[])  
    {  
        int a=10,b=2,c;  
        c=a*b;  
        System.out.print ("Mul="+c);  
    }  
}
```

**P4) Write a program to sub two values.**

```
class atul  
{  
    public static void main(String s[])  
    {  
        int a=10,b=2,c;  
        c=a-b;  
        System.out.print("Sub="+c);  
    }  
}
```

**P5) Write a program to divide two values.**



```

class atul
{
    public static void main(String s[])
    {
        int a=10,b=2;
        float c;
        c=a/b;
        System.out.print ("Div="+c);
    }
}

```

**P6) Write a program to add two value input by user.**

```

import java.io.*;
class atul
{
    void add()throws Exception
    {
        int a,b,c;
        System.out.print("Enter two value=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        a=Integer.parseInt(br.readLine());
        b=Integer.parseInt(br.readLine());
        c=a+b;
        System.out.print("Add:-"+c);
    }
    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
        ob.add();
    }
}

```



**P7) Write a program to sub two value input by user.**

```
import java.io.*;

class atul
{
    void sub()throws Exception
    {
        int a,b,c;
        System.out.print("Enter two value=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        a=Integer.parseInt(br.readLine());
        b=Integer.parseInt(br.readLine());
        c=a-b;
        System.out.print("Sub:- "+c);
    }
    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
        ob.sub();
    }
}
```

**P8) Write a program to mul two values value input by the user.**

```
import java.io.*;

class atul
{
    void mul()throws Exception
    {
        int a,b,c;
        System.out.print("Enter two value=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        a=Integer.parseInt(br.readLine());
```

```

        b=Integer.parseInt(br.readLine());
        c=a*b;
        System.out.print("Mul:-"+c);
    }
    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
        ob.mul();
    }
}

```

**P9) Write a program to divide two value values input by the user.**

```

import java.io.*;
class atul
{
    void div()throws Exception
    {
        int a,b,c;
        System.out.print("Enter two value=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        a=Integer.parseInt(br.readLine());
        b=Integer.parseInt(br.readLine());
        c=a/b;
        System.out.print("div:-"+c);
    }
    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
        ob.div();
    }
}

```

**P10) Write a program to calculate the area to circle.**

```
import java.io.*;

class atul
{
    void area()throws Exception
    {
        int rad;
        float area;
        System.out.print("Enter a rad=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        rad= Integer.parseInt(br.readLine());
        area=3.14*rad*rad;
        System.out.print("Area of Circle="+area);
    }
    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
        ob. area();
    }
}
```

**P11) Write a program to calculate the area of Rect.**

```
import java.io.*;

class atul
{
    void area()throws Exception
    {
        int height,width,area;
        System.out.print("Enter the Height=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        height= Integer.parseInt(br.readLine());
```

```

        System.out.print("Enter the width=");
        width= Integer.parseInt(br.readLine());
        area=height*width;
        System.out.print("Area of Rec="+area);
    }

    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
        ob. area();
    }
}

```

**P12) Write a program to calculate the area of square.**

```

import java.io.*;

class atul
{
    void area()throws Exception
    {
        int rad,area;

        System.out.print("Enter the rad=");

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        rad=Integer.parseInt(br.readLine());
        area=rad*rad;

        System.out.print("Area of squ="+area);
    }

    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
        ob. area();
    }
}

```

**P13) Write a program to find ASCII value of any character.**

```
import java.io.*;

class atul
{
    void ass()throws Exception
    {
        char ch;

        System.out.print("Enter any character=");

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        ch=(char)br.read();

        System.out.print("The ASCII value of any character="+&ch);

    }

    public static void main(String s[])throws Exception
    {
        atul ob=new atul();

        ob. ass();

    }
}
```

**P14) Write a program to Compute Quotient and Remainder.**

```
import java.io.*;

class atul
{
    void ass()throws Exception
    {
        int val1,val2,Quotient,Remainder;

        System.out.print("Enter two values:-");

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        val1=Integer.parseInt(br.readLine());

        val2=Integer.parseInt(br.readLine());

        Quotient=val1/val2;
```

```

        System.out.print("Quotient=%d",Quotient);

        Remainder=val1%val2;

        System.out.print("Remainder=%d",Remainder);

    }

    public static void main(String s[])throws Exception
    {

        atul ob=new atul();

        ob. ass();

    }

}

```

**P15) Write a program to find the size of a variable.**

```

import java.io.*;

class atul
{

    void ass()throws Exception
    {

        int a;

        float b;

        char c;

        double d;

        System.out.print("Size of int data type:-"+sizeof(a)+"\n");

        System.out.print("Size of float data type:-"+sizeof(b) +"\n");

        System.out.print("Size of char data type:-"+sizeof(c) +"\n");

        System.out.print("Size of double data type:-"+sizeof(d) +"\n");

    }

    public static void main(String s[])throws Exception
    {

        atul ob=new atul();

        ob. ass();

    }

}

```

```
}
```

**P16) Write a program to swap two values by using third variable.**

```
import java.io.*;

class atul
{
    void ass()throws Exception
    {
        int a,b,t;

        System.out.print("Enter two values=");

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        a=Integer.parseInt(br.readLine());

        b=Integer.parseInt(br.readLine());

        System.out.print("Before swap the value of a="+a+"and the value of b="+b);

        t=a;

        a=b;

        b=t;

        System.out.print("After swap the value of a="+a+"and the value of b=%d"+b);

    }

    public static void main(String s[])throws Exception
    {

        atul ob=new atul();

        ob. ass();

    }

}
```

**P17) Write a program to swap two value without using third variable.**

```
import java.io.*;

class atul
{
    void ass()throws Exception
```



```

{
    int a,b;
    System.out.print("Enter two values=");
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    a=Integer.parseInt(br.readLine());
    b=Integer.parseInt(br.readLine());
    System.out.print("Before swap the value of a="+a+"and the value of b="+b);
    a=a+b;
    b=a-b;
    a=a-b;
    System.out.print("After swap the value of a="+a+"and the value of b=%d"+b);
}
public static void main(String s[])throws Exception
{
    atul ob=new atul();
    ob. ass();
}
}

```

## Operators

An operator is a symbol that operates on a value or a variable. For example: `+` is an operator to perform addition.

C has a wide range of operators to perform various operations.

### 1) Arithmetic Operators

+	Add two or more values
-	Sub two or more values
*	mul two or more values
/	div two or more values
%	Remainder of two values

**P18) Write a program to perform a Arithmetic Operators.**

```

import java.io.*;

class atul
{
    void ass()throws Exception
    {
        int a,b,add,sub,mul,div,rem;

        System.out.print("Enter two values=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        a=Integer.parseInt(br.readLine());
        b=Integer.parseInt(br.readLine());

        add=a+b;
        sub=a-b;
        mul=a*b;
        div=a/b;
        rem=a%b;

        System.out.print("Add of two value="+add);
        System.out.print("\nSub of two value="+sub);
        System.out.print("\nMul of two value="+mul);
        System.out.print("\nDiv of two value="+div);
        System.out.print("\nRem of two value="+rem);
    }

    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
        ob. ass();
    }
}

```

**2) Assignment Operators**

```

=      Assign a value
+=
-=
*=
/=

```

**P19) Write a program to perform Assignment Operators.**

```

import java.io.*;

class atul
{
    void ass()throws Exception
    {
        int n,a;
        printf("Enter any value=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());

        a=n;
        System.out.print("Assign a value to a="+a);

        n+=2;

        System.out.print("\nThe value of n increment by 2 after increment the
value of n="+n);

        n-=3;

        System.out.print("\nThe value of n decrement by 3 after decrement the
value of n="+n);

        n*=2;

        System.out.print("\nThe value of n multiply by 2 after multiply the value of
n="+n);

        n/=2;

        System.out.print("\nThe value of n divide by 2 after dividation the value of
n="+n);
    }

    public static void main(String s[])throws Exception
    {

```

```

        atul ob=new atul();
        ob. ass();
    }
}

```

### **3)Relational Operators**

==	comp two values
>	Grater then
<	Less then
>=	Grater then and equal to
<=	Less then and equal to
!=	not equal to

**P20) Write a program to define Relational Operators.**

```

import java.io.*;
class atul
{
    void ass()throws Exception
    {
        int a,b;
        System.out.print("Enter two values=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        a=Integer.parseInt(br.readLine());
        b=Integer.parseInt(br.readLine());
        System.out.print("1 means true and 0 means false");
        System.out.print("\nThe value of a and b are equal or not" +(a==b));
        System.out.print("\nThe value of a and b are grater then or not" +(a>b));
        System.out.print("\nThe value of a and b are less then or not" +(a<b));
        System.out.print("\nThe value of a and b are grater then equal to or not
"+(a>=b));
        System.out.print("\nThe value of a and b are less the equal to or not
"+(a<=b));
        System.out.print("\nThe value of a and b are not equal" +(a!=b));
    }
}

```

```

    }

    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
        ob. ass();
    }
}

```

#### **4)Logical Operators**

&&      And operators

||        or operator

And(&&) operator:-

Condition 1	Condition 2	Result
T	T	T
T	F	F
F	T	F
F	F	F

Or (||) Operators:-

Condition 1	Condition 2	Result
T	T	T
T	F	T
F	T	T
F	F	F

## Controlling Statements

### Types of Controlling Statements

- 1) If
- 2) If else
- 3) Nested if else
- 4) Else if
- 5) Switch

#### **1)if Statement:-**

Syntax:-

```

    if(condition)
    {
        Statements;
    }

```

**P21) Write a program to check that the given value is greater than 10 print your name.**

```

import java.io.*;

class atul
{
    void ass()throws Exception
    {
        int n;

        System.out.print("Enter any value=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());

        if(n>10)
        {
            System.out.print("Atul");
        }
    }

    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
    }
}

```

```

        ob. ass();
    }
}

```

## 2) if else Statements:-

Syntax:-

```

        If(condition)
        {
            Statements;
        }
        else
        {
            Statements;
        }

```

**P22) write a program to check that the give value is even and odd.**

```

import java.io.*;

class atul
{
    void ass()throws Exception
    {
        int n;
        System.out.print("Enter any value=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());
        if(n%2==0)
        {
            System.out.print("Value is even");
        }
        else
        {
            System.out.print("value is odd");
        }
    }
}

```



```

    }
}
public static void main(String s[])throws Exception
{
    atul ob=new atul();
    ob. ass();
}
}

```

**P23) Write a program to check that the give character is vow or cons.**

```

import java.io.*;
class atul
{
    void ass()throws Exception
    {
        char ch;
        System.out.print("Enter any character=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        ch=(char)br.read();
        if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')
        {
            System.out.print("The given char is vow");
        }
        else
        {
            System.out.print("The given char is con");
        }
    }
    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
    }
}

```

```

        ob. ass();
    }
}

```

**P24) write a program if the sugar rate is grater then 60 print ½ kg other wise 1 kg.**

```

import java.io.*;
class atul
{
    void ass()throws Exception
    {
        int n;
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());
        if(n>60)
        {
            System.out.print("1/2 KG");
        }
        else
        {
            System.out.print("1 KG");
        }
    }
    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
        ob. ass();
    }
}

```

**P25) Write a program to check that the grater value b/w two variables.**

```

import java.io.*;
class atul

```

```

{
    void ass()throws Exception
    {
        int val1,val2;
        System.out.print("Enter two values=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        val1=Integer.parseInt(br.readLine());
        val2=Integer.parseInt(br.readLine());
        if(val1>val2)
        {
            System.out.print("Value 1 is Grater");
        }
        else
        {
            System.out.print("Value 2 is Grater");
        }
    }
    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
        ob. ass();
    }
}

```

### 3) Nested if else:-

Syntax:-

```

if(condition)
{
    If(condition)
    {
        If(condition)
    }
}

```

```

        {
            Statements;
        }
    else
    {
        Statements;
    }
}
else
{
    Statements;
}
}
else
{
    If(condition)
    {
        Statements;
    }
    else
    {
        Statements;
    }
}

```

**P26) Write a program to check that the grate value b/w three variables.**

```

import java.io.*;
class atul
{
    void ass()throws Exception
    {

```

```
int val1,val2,val3;

System.out.print("Enter three values:-");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

val1=Integer.parseInt(br.readLine());

val2=Integer.parseInt(br.readLine());

val3=Integer.parseInt(br.readLine());

if(val1>val2)
{
    if(val1>val3)
    {
        System.out.print("Value 1 is grater");
    }
    else
    {
        System.out.print("Value 3 is grater");
    }
}
else
{
    if(val2>val3)
    {
        System.out.print("Value 2 is grater");
    }
    else
    {
        System.out.print("Value 3 is grater");
    }
}
}
```

```
public static void main(String s[])throws Exception
{
    atul ob=new atul();
    ob. ass();
}
}
```

#### 4)else if Statements:-

Syntax:-

```
if(condition)
{
    Statements;
}
else if(condition)
{
    Statements;
}
else if(condition)
{
    Statements;
}
else if(condition)
{
    Statements;
}
.
.
.
else
```

```
{
    Statements;
}
```

**P27) Write a program to enter a week no and then print the related week day.**

```
import java.io.*;
class atul
{
    void ass()throws Exception
    {
        int n;
        System.out.print("Enter a week no:-");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());
        if(n==1)
        {
            System.out.print("Sunday");
        }
        else if(n==2)
        {
            System.out.print("Monday");
        }
        else if(n==3)
        {
            System.out.print("Tuesday");
        }
        else if(n==4)
        {
            System.out.print("Wednesday");
        }
    }
}
```



```

        else if(n==5)
        {
            System.out.print("Thuresday");
        }
        else if(n==6)
        {
            System.out.print("Friday");
        }
        else if(n==7)
        {
            System.out.print("Saturday");
        }
        else
        {
            System.out.print("!Wrong Week No");
        }
    }

    public static void main(String s[])throws Exception
    {
        atul ob=new atul();
        ob. ass();
    }
}

```

**P28) Write a program to enter a month no and then print the month name.**

```

import java.io.*;

class atul
{
    void ass()throws Exception
    {
        int n;
    }
}

```

```
System.out.print("Enter a week no:-");  
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  
n=Integer.parseInt(br.readLine());  
if(n==1)  
{  
    System.out.print("Jan");  
}  
else if(n==2)  
{  
    System.out.print("Feb");  
}  
else if(n==3)  
{  
    System.out.print("Mar");  
}  
else if(n==4)  
{  
    System.out.print("Apr");  
}  
else if(n==5)  
{  
    System.out.print("May");  
}  
else if(n==6)  
{  
    System.out.print("Jun");  
}  
else if(n==7)  
{
```

```
        System.out.print("july");
    }
    else if(n==8)
    {
        System.out.print("Aug");
    }
    else if(n==9)
    {
        System.out.print("Sep");
    }
    else if(n==10)
    {
        System.out.print("Oct");
    }
    else if(n==11)
    {
        System.out.print("Nov");
    }
    else if(n==12)
    {
        System.out.print("Dec");
    }
    else
    {
        System.out.print("!Wrong Month no.");
    }
}

public static void main(String s[])throws Exception
{
```

```

        atul ob=new atul();
        ob. ass();
    }
}

```

**P29) Write a program to create a simple calculator by using else if.**

```

import java.io.*;

class atul
{
    void ass()throws Exception
    {
        int n,a,b,c;
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.print("1.Add\n2.Sub\n3.Mul\n4.Div\n");
        System.out.print("Enter your choice=");
        n=Integer.parseInt(br.readLine());
        System.out.print("Enter two value=");
        a=Integer.parseInt(br.readLine());
        b=Integer.parseInt(br.readLine());
        if(n==1)
        {
            c=a+b;
            System.out.print("Sum="+c);
        }
        else if(n==2)
        {
            c=a-b;
            System.out.print("Sub="+c);
        }
        else if(n==3)
        {

```

```
        c=a*b;
        System.out.print("Mul="+c);
    }
    else if(n==4)
    {
        c=a/b;
        System.out.print("Div="+c);
    }
    else
    {
        System.out.print("!Wrong Choice");
    }
}
public static void main(String s[])throws Exception
{
    atul ob=new atul();
    ob. ass();
}
}
```

#### 5)Switch Case:-

Syntax:-

```
switch(vari)
{
    case 1:  Statements;
            break;
    case 2:  Statements;
            break;
    case 3:  Statements;
            break;
```

```

        case 4: Statements;
                break;

        .
        .
        .
        .
        .

        default :  statements;

    }

```

**P30) Write a program to enter a week no and print the related week name.**

```

import java.io.*;

class atul
{
    void ass()throws Exception
    {
        int n;

        System.out.print("Enter a week no:-");

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        n=Integer.parseInt(br.readLine());

        switch(n)
        {

            case 1: System.out.print("Sunday");

                    break;

            case 2: System.out.print("Monday");

                    break;

            case 3: System.out.print("Tuesday");

                    break;

            case 4: System.out.print("Wednesday");

                    break;

```

```

        case 5: System.out.print("Thuesday");
                break;
        case 6: System.out.print("Friday");
                break;
        case 7: System.out.print("Saturday");
                break;
        default: System.out.print("!Wrong choice");
    }
}

public static void main(String s[])throws Exception
{
    atul ob=new atul();
    ob. ass();
}
}

```

**P31) Write a program to enter a month no. and the print a related month name.**

```

import java.io.*;

class atul
{
    void ass()throws Exception
    {
        int n;

        System.out.print("Enter a week no:-");

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());

        switch(n)
        {
            case 1: System.out.print("Jan");
                    break;
            case 2: System.out.print("Feb");

```



```
        break;
    case 3: System.out.print("mar");
        break;
    case 4: System.out.print("Apr");
        break;
    case 5: System.out.print("May");
        break;
    case 6: System.out.print("Jun");
        break;
    case 7: System.out.print("july");
        break;
    case 8: System.out.print("Aug");
        break;
    case 9: System.out.print("sep");
        break;
    case 10: System.out.print("Oct");
        break;
    case 11: System.out.print("Nov");
        break;
    case 12: System.out.print("Dec");
        break;
    default: System.out.print("! Wrong Choice");
}
}
public static void main(String s[])throws Exception
{
    atul ob=new atul();
    ob. ass();
}
```

```
}
```

**P32) Write a program to create a simple calculator by using switch.**

```
import java.io.*;

class atul
{
    void ass()throws Exception
    {
        int n,a,b,c;

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.print("1.Add\n2.Sub\n3.Mul\n4.Div\n");
        System.out.print("Enter your choice=");
        n=Integer.parseInt(br.readLine());
        System.out.print("Enter two value=");
        a=Integer.parseInt(br.readLine());
        b=Integer.parseInt(br.readLine());
        switch(n)
        {
            case 1: c=a+b;
                    System.out.print("Sum="+c);
                    break;
            case 2: c=a-b;
                    System.out.print("Sub="+c);
                    break;
            case 3: c=a*b;
                    System.out.print("Mul="+c);
                    break;
            case 4: c=a/b;
                    System.out.print("Div="+c);
                    break;
```

```

        default: System.out.print("!Wrong Choice");
    }
}

public static void main(String s[])throws Exception
{
    atul ob=new atul();
    ob. ass();
}
}

```

## Loops

In programming language, looping means to execute a set of statements for a specified number of times or till a condition remains true. In a loop, the compiler checks for the condition specified for executing the loop and continues executing the statements mentioned within the loop till the condition remains true. The execution of the loop stops as soon as the condition becomes false.

Re-execution of a statement is called loops while the condition remains true.

### Types of loop

- 1)for loop
- 2)nested for loop
- 3)while loop
- 4)do while loop

#### **1)for loop:-**

##### **Syntax:-**

```

for(initialization, condition, Inc/Dec)
{
    Statements;
}

```

**P33) Write a program to print your name 10 times.**

```

class atul
{
    void ass()

```

```

    {
        for(int i=1;i<=10;i++)
        {
            System.out.print("Atul Kumar Varshney\n");
        }
    }
    public static void main(String s[])
    {
        atul ob=new atul();
        ob. ass();
    }
}

```

**P34) Write a program to print 1 to 10.**

```

class atul
{
    void ass()
    {
        for(int i=1;i<=10;i++)
        {
            System.out.println(i);
        }
    }
    public static void main(String s[])
    {
        atul ob=new atul();
        ob. ass();
    }
}

```

**P35) Write a program to print a to z.**

```

class atul

```

```

{
    void ass()
    {
        for(char i='a';i<='z';i++)
        {
            System.out.print("\t"+i);
        }
    }
    public static void main(String s[])
    {
        atul ob=new atul();
        ob. ass();
    }
}

```

**P36) Write a program to print A to Z.**

```

class atul
{
    void ass()
    {
        for(char i='a';i<='z';i++)
        {
            System.out.print("\t"+i);
        }
    }
    public static void main(String s[])
    {
        atul ob=new atul();
        ob. ass();
    }
}

```

**P36) Write a program to print A to Z.**

```
class atul
{
    void ass()
    {
        for(char i='A';i<='Z';i++)
        {
            System.out.print("\t"+i);
        }
    }
    public static void main(String s[])
    {
        atul ob=new atul();
        ob. ass();
    }
}
```

**P37) Write a program to print a Fibonacci sequence.**

```
class atul
{
    void ass()throws Exception
    {
        int a=0,b=1,n,c;
        System.out.print("Enter the limit=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());
        if(n>2)
        {
            System.out.print("%d\t%d\t",a,b);
            for(int i=3;i<=n;i++)
            {
```

```

        c=a+b;
        System.out.print("%d\t",c);
        a=b;
        b=c;
    }
}
else
{
    System.out.print("The value is grater the 2");
}
}
public static void main(String s[])throws Exception
{
    atul ob=new atul();
    ob. ass();
}
}

```

## 2)Nested for Loop:-

### Syntax:-

```

for(initialization,condition,Inc/Dec)
{
    for(initialization,condition,Inc/Dec)
    {
        Statements;
    }
    for(initialization,condition,Inc/Dec)
    {
        Statements;
    }
}

```

```
}
```

**P38) Write a program to print the pattern**

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

import java.io.*;
class atul
{
    void ass()throws Exception
    {
        int n;
        System.out.print ("Enter the limit=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=i;j++)
            {
                System.out.print("*");
            }
            System.out.print("\n");
        }
    }
    public static void main(String s[]) throws Exception
    {
        atul ob=new atul();
    }
}
```



```

        ob. ass();
    }
}

```

**P39) Write a program to print the pattern**

```
1
```

```
12
```

```
123
```

```
1234
```

```
12345
```

```
123456
```

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

```
class atul
```

```
{
```

```
    void ass()throws Exception
```

```
    {
```

```
        int n;
```

```
        System.out.print("Enter the limit=");
```

```
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
```

```
        n=Integer.parseInt(br.readLine());
```

```
        for(int i=1;i<=n;i++)
```

```
        {
```

```
            for(int j=1;j<=i;j++)
```

```
            {
```

```
                System.out.print(j);
```

```
            }
```

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

```
        }
```

```
    }
```

```
    public static void main(String s[]) throws Exception
```

```
    {
```

```

        atul ob=new atul();
        ob. ass();
    }
}

```

**P40) Write a program to print the pattern**

```

    *
  * *
 ***
****
*****

import java.io.*;
class atul
{
    void ass()throws Exception
    {
        int n;
        System.out.print("Enter the limit=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=(n-i);j++)
            {
                System.out.print(" ");
            }
            for(int k=1;k<=i;k++)
            {
                System.out.print("*");
            }
            System.out.print("\n");
        }
    }
}

```

```

        }
    }
    public static void main(String s[]) throws Exception
    {
        atul ob=new atul();
        ob. ass();
    }
}

```

**P41) Write a program to print the pattern**

```

1
12
123
1234
12345
123456

```

```

import java.io.*;
class atul
{
    void ass()throws Exception
    {
        int n;
        System.out.print("Enter the limit=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=(n-i);j++)
            {
                System.out.print(" ");
            }
        }
    }
}

```

```

        for(int k=1;k<=i;k++)
        {
            System.out.print("%d",k);

        }

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

    }

}

public static void main(String s[]) throws Exception
{
    atul ob=new atul();

    ob. ass();

}

}

```

**P42) Write a program to print the pattern**

```
*                                     *
```

```
**                               **
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
import java.io.*;

class atul

{

    void ass()throws Exception

    {

        int n;

        System.out.print("Enter the limit=");

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        n=Integer.parseInt(br.readLine());

        for(int i=1;i<=n;i++)
```

```

        {
            for(int j=1;j<=i;j++)
            {
                System.out.print("*");
            }
            for(int l=1;l<=(n*2)-(i*2);l++)
            {
                System.out.print(" ");
            }
            for(int k=1;k<=i;k++)
            {
                System.out.print("*");
            }
            System.out.print("\n");
        }
    }

    public static void main(String s[]) throws Exception
    {
        atul ob=new atul();
        ob. ass();
    }
}

```

**P43) Write a program to print the pattern**

```

1          1
12         21
123        321
1234       4321
12345      54321
123456654321

import java.io.*;

```

```
class atul
{
    void ass()throws Exception
    {
        int n;
        System.out.print("Enter the limit=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=i;j++)
            {
                System.out.print(j);
            }
            for(int l=1;l<=(n*2)-(i*2);l++)
            {
                System.out.print(" ");
            }
            for(int k=i;k>=1;k--)
            {
                System.out.print(k);
            }
            System.out.print("\n");
        }
    }
    public static void main(String s[]) throws Exception
    {
        atul ob=new atul();
        ob. ass();
    }
}
```

```
}
```

**P44) Write a program to print the pattern**

```
*****
```

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

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

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

```
**       **
```

```
*        *
```

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

```
class atul
```

```
{
```

```
    void ass( ) throws Exception
```

```
    {
```

```
        int n;
```

```
        System.out.print("Enter the limit=");
```

```
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
```

```
        n=Integer.parseInt(br.readLine());
```

```
        for(int i=1;i<=n;i++)
```

```
        {
```

```
            for(int j=1;j<=(n+1)-i;j++)
```

```
            {
```

```
                System.out.print("*");
```

```
            }
```

```
            for(int l=1;l<=a;l++)
```

```
            {
```

```
                System.out.print(" ");
```

```
            }
```

```
            a=a+2;
```

```
            for(int k=1;k<=(n+1)-i;k++)
```

```

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

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

**P45) Write a program to print**

```

*
***
*****
*****
*****
*****
*****
*****
***
*

import java.io.*;
class atul
{
    void ass( ) throws Exception
    {
        for(int i=1;i<=5;i++)
        {
            for(int j=1;j<=i;j++)

```



```

        {
            System.out.print("*");
        }
        System.out.print("\n");
    }
    for(int i=1;i<=4;i++)
    {
        for(int j=1;j<=(5-i);j++)
        {
            System.out.print("*");
        }
        System.out.print("\n");
    }
}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

### 3) while loop:-

#### Syntax:-

```

while(condition)
{
    Statements;
}

```

**P46) Write a program to print your name 10 times.**

```

import java.io.*;

class atul
{

```

```

void ass( ) throws Exception
{
    int i=1;
    while(i<=10)
    {
        System.out.print("Atul\n");
        i++;
    }
}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

**P47) Write a program to print the reverse of any value.**

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int n,rev=0,r;
        System.out.print("Enter any value=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());
        while(n!=0)
        {
            r=n%10;
            rev=(rev*10)+r;
            n=n/10;
        }
    }
}

```

```

    }

    System.out.print("Reverse of any value="+rev);

}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

**P48) Write a program to check that the given value is palindrome or not.**

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int n,t,r,rev=0;

        System.out.print("Enter any value=");

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());

        t=n;
        while(n!=0)
        {
            r=n%10;
            rev=(rev*10)+r;
            n=n/10;
        }

        if(rev==t)
        {
            System.out.print("Value is palindrome");
        }
    }
}

```

```

        else
        {
            System.out.print("Value is not palindrome");
        }
    }

    public static void main(String s[ ]) throws Exception
    {
        atul ob=new atul( );
        ob. ass( );
    }
}

```

**P49) Write a program to check that the given value is Armstrong or not.**

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int n,r,t,arm=0;
        System.out.print("Enter any value=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());
        t=n;
        while(n!=0)
        {
            r=n%10;
            arm=(r*r*r)+arm;
            n=n/10;
        }
        if(arm==t)
        {

```

```

        System.out.print("Value is Armstrong");
    }
    else
    {
        System.out.print("Value is not Armstrong");
    }
}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

#### 4)do while:-

##### Syntax:-

```

do
{
    Statements;
}while(condition);

```

#### Difference between While and do while loop.

While	Do while
In While loop the controlling condition appears at start of the loop.	In do while loop the controlling condition appears at end of the loop.
The statements do not execute if the condition is false.	The statements are executed at least one time if the condition is true as well as false.
Entry controlled loop.	Exit controlled loop.

#### P50) Write a program to create a simple calculator in re-execution mode if we press y.

```

import java.io.*;

class atul
{
    void ass( ) throws Exception

```

```

{
    int a,b,c,n;
    char ch='n';
    do
    {
        System.out.print("1.Add\n2.Sub\n3.Mul\n4.Div\n");
        System.out.print("Enter your choice=");
        BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());
        System.out.print("Enter two values=");
        a=Integer.parseInt(br.readLine());
        b=Integer.parseInt(br.readLine());
        switch(n)
        {
            case 1: c=a+b;
                System.out.print("Sum="+c);
                break;
            case 2: c=a-b;
                System.out.print("Sub="+c);
                break;
            case 3: c=a*b;
                System.out.print("Mul="+c);
                break;
            case 4: c=a/b;
                System.out.print("Div="+c);
                break;
            default: System.out.print("!Wrong Choice");
        }
        System.out.print("\nDo you want to cont press y=");
    }
}

```

```

        ch=(char)br.read();
    }while(ch=='y');

}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

## Break Statement

The break statement is one of the loop control statement that terminates the loop. C programming language uses the keyword 'break' for using the break statement. In other words, whenever the compiler encounters the break statement it terminates the loop.

### P51) W.A.P. to show the break statement.

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int n;

        System.out.print("Enter the value to break a loop:-");

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        n=Integer.parseInt(br.readLine());

        for(int i=1;i<=100;i++)
        {
            if(i==n)
            {
                break;
            }
        }
    }
}

```

```

        System.out.print("Atul\n"+i);
    }
}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

## Continue Statement

The continue statement is the next loop control statement that continues to execute the loop. C programming language use the keyword 'continue' for using the continue statement. In other words, whenever the continue statement arrives the compiler immediately skips the current statements of the loop and starts the next iteration of the loop.

**P52) W.A.P. to show the continue statement it exc. the 2 value after printing the value.**

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        for(int i=1;i<=10;i++)
        {
            if(i==2)
            {
                continue;
            }
            System.out.print("\n"+i);
        }
    }

    public static void main(String s[ ]) throws Exception
    {

```



```

        atul ob=new atul( );
        ob. ass();
    }
}

```

## Array

An array is a type of data structure that can hold multiple values of similar type of data. According to the technical definition of the array, "Array is a homogeneous data structure that sequentially stores the same types of operations on these data".

Array can store similar data type. Each data item of an array is called the array element. For Ex:- if the data type of the array is char then it only stores the character data or string.

### Types of an Array

1)1-D Array

2)2-D Array

#### 1)1-D Array:-

##### Array Declaration:-

##### Syntax:-

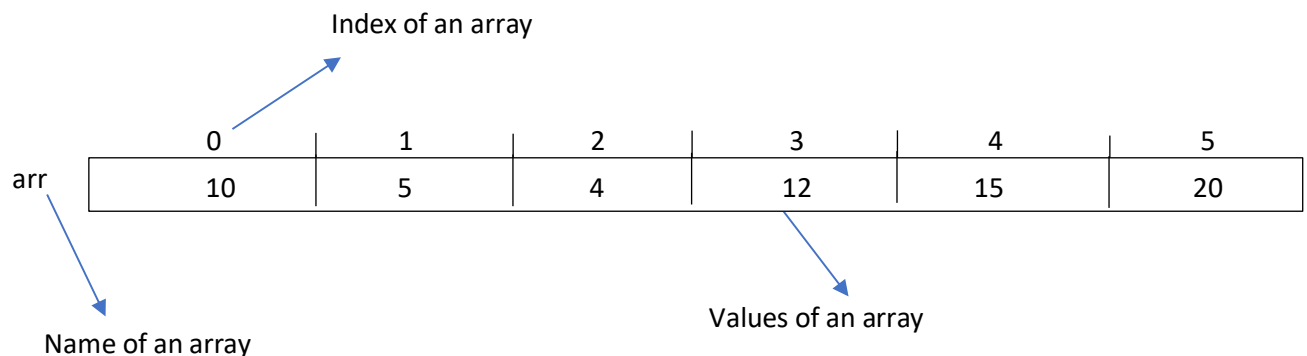
Data\_type name\_of\_array[]=new Data\_type [Size];

Ex:- int arr[]=new int[6];

Declares an integer array where the size of the array is 6. The name of the array is arr. It can store any 6 numbers that would be stored from location 0 to 6 in the computer memory.

The index no. of an array start from 0 to size-1.

An array looks like.



##### Initialisation at Compile Time:-

```
int arr[6]={12,13,5,3,1,15};
```

**Initialisation at Run Time:-**

```

int arr[]=new int[6];

System.out.print("Enter values in an array=");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

for(int i=0;i<6;i++)
{
    arr[i]=Integer.parseInt(br.readLine());
}

```

**Accessing Array Elements:-**

```

System.out.print(arr[2]);

```

**P54) W.A.P. to sum of all elements in an 1-D array.**

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int arr[]=new int[6];
        int s=0;
        System.out.print("Enter values in an array=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        for(int i=0;i<6;i++)
        {
            arr[i]= Integer.parseInt(br.readLine());
        }
        for(int i=0;i<6;i++)
        {
            s=s+arr[i];
        }
        System.out.print("Sum of all elements="+s);
    }
}

```

```

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

**P55) W.A.P. to find maximum value in a 1 D Array.**

```

import java.io.*;
class atul
{
    void ass( ) throws Exception
    {
        int arr[]=new int[6];
        int max;
        System.out.print("Enter values in an array=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        for(int i=0;i<6;i++)
        {
            arr[i]= Integer.parseInt(br.readLine());
        }
        max=arr[0];
        for(int i=0;i<6;i++)
        {
            if(max<arr[i])
            {
                max=arr[i];
            }
        }
        System.out.print("Maximum value in an array="+max);
    }
    public static void main(String s[ ]) throws Exception

```

```

    {
        atul ob=new atul( );
        ob. ass( );
    }
}

```

**P56) W.A.P. to find minimum value in a 1-D Array.**

```

import java.io.*;
class atul
{
    void ass( ) throws Exception
    {
        int arr[]=new int[6];
        int min;
        System.out.print("Enter values in an array=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        for(int i=0;i<6;i++)
        {
            arr[i]= Integer.parseInt(br.readLine());
        }
        min=arr[0];
        for(int i=0;i<6;i++)
        {
            if(min>arr[i])
            {
                min=arr[i];
            }
        }
        System.out.print("Minimum value in an array="+min);
    }
    public static void main(String s[ ]) throws Exception

```

```

    {
        atul ob=new atul( );
        ob. ass( );
    }
}

```

**P57) W.A.P. to find any element in a 1-D Array.**

```

import java.io.*;
class atul
{
    void ass( ) throws Exception
    {
        int arr[]=new int[6];
        int n,f=0;
        System.out.print("Enter values in an array=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        for(int i=0;i<6;i++)
        {
            arr[i]= Integer.parseInt(br.readLine());
        }
        System.out.print("What value you will find=");
        n=Integer.parseInt(br.readLine());
        for(int i=0;i<6;i++)
        {
            if(n==arr[i])
            {
                f=1;
            }
        }
        if(f==1)
        {

```

```

        System.out.print("Value present in an array");
    }
    else
    {
        System.out.print("Value not present in an array");
    }
}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

**P58) W.A.P. to sort an array in Ascending Order.**

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int arr[]=new int[6];
        int t;
        System.out.print("Enter values in an array=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        for(int i=0;i<6;i++)
        {
            arr[i]= Integer.parseInt(br.readLine());
        }
        for(int i=0;i<6;i++)
        {
            for(int j=i;j<6;j++)

```

```

        {
            if(arr[i]>arr[j])
            {
                t=arr[i];
                arr[i]=arr[j];
                arr[j]=t;
            }
        }
    }

    System.out.print("The sorted array=");
    for(int i=0;i<6;i++)
    {
        System.out.print("\t"+arr[i]);
    }
}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

**P59) W.A.P. to sort an array in descending order.**

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int arr[]=new int[6];
        int t;
        System.out.print("Enter values in an array=");
    }
}

```

```
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
for(int i=0;i<6;i++)
{
    arr[i]= Integer.parseInt(br.readLine());
}
for(int i=0;i<6;i++)
{
    for(int j=i;j<6;j++)
    {
        if(arr[i]<arr[j])
        {
            t=arr[i];
            arr[i]=arr[j];
            arr[j]=t;
        }
    }
}
System.out.print("The Sorted array=");
for(int i=0;i<6;i++)
{
    System.out.print ("\\t"+arr[i]);
}
}
public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}
```



**P60) W.A.P. to replace a particular index value in 1-D array.**

```
import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int arr[]=new int[6];
        int n,p;
        System.out.print("Enter values in an array=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        for(int i=0;i<6;i++)
        {
            arr[i]= Integer.parseInt(br.readLine());
        }
        System.out.print("Values in an array=");
        for(int i=0;i<6;i++)
        {
            System.out.print("\t"+arr[i]);
        }
        System.out.print("\nWhat value you want to replace=");
        n= Integer.parseInt(br.readLine());
        System.out.print("Enter the index of a value=");
        p= Integer.parseInt(br.readLine());
        arr[p]=n;
        System.out.print("The change array=");
        for(int i=0;i<6;i++)
        {
            System.out.print("\t"+arr[i]);
        }
    }
}
```

```

    }

    public static void main(String s[ ]) throws Exception
    {
        atul ob=new atul( );
        ob. ass( );
    }
}

```

**P61) W.A.P. to replace value if the array element is even replace 0 if the array element is odd replace 1.**

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int arr[]=new int[6];
        System.out.print("Enter values in an array=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        for(int i=0;i<6;i++)
        {
            arr[i]= Integer.parseInt(br.readLine());
        }
        for(int i=0;i<6;i++)
        {
            if(arr[i]%2==0)
            {
                arr[i]=0;
            }
            else
            {
                arr[i]=1;
            }
        }
    }
}

```

```

        }
    }
    System.out.print("The Change array=");
    for(int i=0;i<6;i++)
    {
        System.out.print("\t"+arr[i]);
    }
}
public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

**P62) W.A.P. to Add two array.**

```

import java.io.*;
class atul
{
    void ass( ) throws Exception
    {
        int arr[]=new int[6];
        int arr1[]=new int[6];
        int sum[]=new int[6];
        System.out.print("Enter values in an array=");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        for(int i=0;i<6;i++)
        {
            arr[i]= Integer.parseInt(br.readLine());
        }
        System.out.print("Enter values in second array=");
    }
}

```

```

        for(int i=0;i<6;i++)
        {
            arr[i]= Integer.parseInt(br.readLine());
        }
        for(int i=0;i<6;i++)
        {
            sum[i]=arr[i]+arr[i];
        }
        System.out.print("The add array=");
        for(int i=0;i<7;i++)
        {
            System.out.print("\t"+sum[i]);
        }
    }
    public static void main(String s[ ]) throws Exception
    {
        atul ob=new atul( );
        ob. ass( );
    }
}

```

## **2)2-D Array:-**

If the array contains two dimensions then it is called the two- dimensional array or 2-D array. This 2D-array is also called the matrix. Matrix is a combination of the row and columns.

### **2-D Array Declaration:-**

**Syntax:-** data\_type name\_of array[][]=new data\_type[row][columns];

**Ex:-** int arr[][]=new int [3][4];

The following statement declares and initializes an integer 2-D array where the row size and column size of the array are 3 and 4 respectively.

### **An 2-D array looks like:-**

**Initialization at Compile Time:-**

```
int arr[3][4]={1,2,3,4},{5,6,7,8},{9,10,11,12}};
```

**Initialization at Run Time:-**

```
int arr[][]=new int[3][4];

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter values in an array=");

for(int i=0;i<3;i++)
{
    for(int j=0;j<4;j++)
    {
        arr[i][j]=Integer.parseInt(br.readLine());
    }
}
```

**Accessing array Elements:-**

```
System.out.print(arr[2][2]);
```

**P63) W.A.P. to sum of all values in 2-D array.**

```
import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int arr[][]=new int[3][3];

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        System.out.print("Enter values in an array=");

        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                arr[i][j]=Integer.parseInt(br.readLine());
            }
        }
    }
}
```

```

        }
    }
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            s=s+arr[i][j];
        }
    }
    System.out.print("Sum of all values="+s);
}
public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

**P64)W.A.P. to find maximum no. in 2-D array.**

```

import java.io.*;
class atul
{
    void ass( ) throws Exception
    {
        int arr[][]=new int[3][3];
        int max;
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter values in an array=");
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)

```

```

        {
            arr[i][j]=Integer.parseInt(br.readLine());
        }
    }
    max=arr[0][0];
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            if(max<arr[i][j])
            {
                max=arr[i][j];
            }
        }
    }
    System.out.print("Maximum value in an array=%d",max);
}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

**P65) W.A.P. to find minimum value in 2-D array.**

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int arr[][]=new int[3][3];
    }
}

```

```

int min;

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter values in an array=");

for(int i=0;i<3;i++)
{
    for(int j=0;j<3;j++)
    {
        arr[i][j]=Integer.parseInt(br.readLine());
    }
}

min=arr[0][0];

for(int i=0;i<3;i++)
{
    for(int j=0;j<3;j++)
    {
        if(min>arr[i][j])
        {
            min=arr[i][j];
        }
    }
}

System.out.print("Minimum value in an array="+min);

}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

**P66) W.A.P. to add all diagonal element in 2-D array.**



```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int arr[][]=new int[3][3];

        int s=0;

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        System.out.print("Enter values in an array=");

        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                arr[i][j]=Integer.parseInt(br.readLine());
            }
        }

        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                if(i==j)
                {
                    s=s+arr[i][j];
                }
            }
        }

        System.out.print("Add all diagonal="+s);
    }

    public static void main(String s[ ]) throws Exception
    {

```

```

       atul ob=new atul( );
        ob. ass( );
    }
}

```

**P67)W.A.P. to find maximum in diagonal in 2-D array.**

```

import java.io.*;
class atul
{
    void ass( ) throws Exception
    {
        int arr[][]=new int[3][3];
        int max;
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter values in an array=");
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                arr[i][j]=Integer.parseInt(br.readLine());
            }
        }
        max=arr[0][0];
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                if(i==j)
                {
                    if(max<arr[i][j])
                    {

```

```

                                max=arr[i][j];
                                }
                                }
                                }

                                }

                                System.out.print("Maximum value in diagonal="+max);
                                }

                                public static void main(String s[ ]) throws Exception
                                {

                                    atul ob=new atul( );
                                    ob. ass( );

                                }

                                }

```

**P68)W.A.P. to find minimum value in diagonal.**

```

import java.io.*;

class atul
{

    void ass( ) throws Exception
    {

        int arr[][]=new int[3][3];

        int min;

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        System.out.print("Enter values in an array=");

        for(int i=0;i<3;i++)
        {

            for(int j=0;j<3;j++)
            {

                arr[i][j]=Integer.parseInt(br.readLine());

            }

        }

    }

}

```

```

        min=arr[0][0];
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                if(i==j)
                {
                    if(min>arr[i][j])
                    {
                        min=arr[i][j];
                    }
                }
            }
        }
        System.out.print("Minimum value in an array="+min);
    }
    public static void main(String s[ ]) throws Exception
    {
        atul ob=new atul( );
        ob. ass( );
    }
}

```

**P69)W.A.P. to find maximum value in lower triangular.**

```

import java.io.*;
class atul
{
    void ass( ) throws Exception
    {
        int arr[][]=new int[3][3];
        int max;
    }
}

```

```

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.print("Enter values in an array=");
for(int i=0;i<3;i++)
{
    for(int j=0;j<3;j++)
    {
        arr[i][j]=Integer.parseInt(br.readLine());
    }
}
max=arr[1][0];
for(int i=0;i<3;i++)
{
    for(int j=0;j<3;j++)
    {
        if(i>j)
        {
            if(max<arr[i][j])
            {
                max=arr[i][j];
            }
        }
    }
}
System.out.print("Maximum in lower triangular="+max);
}
public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}

```

```

    }
}

```

**P70) W.A.P. to find minimum in value in lower triangular.**

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int arr[][]=new int[3][3];
        int min;

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter values in an array=");
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                arr[i][j]=Integer.parseInt(br.readLine());
            }
        }
        min=arr[1][0];
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                if(i>j)
                {
                    if(min>arr[i][j])
                    {
                        min=arr[i][j];
                    }
                }
            }
        }
    }
}

```

```

        }
    }
}

System.out.print("Minimum value in an lower triangular="+min);

}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

**P71) W.A.P. to find maximum value in upper triangular.**

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int arr[][]=new int[3][3];
        int max;

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter values in an array=");
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                arr[i][j]=Integer.parseInt(br.readLine());
            }
        }

        max=arr[0][1];
        for(int i=0;i<3;i++)
    }
}

```

```

        {
            for(int j=0;j<3;j++)
            {
                if(i<j)
                {
                    if(max<arr[i][j])
                    {
                        max=arr[i][j];
                    }
                }
            }
        }

        System.out.print("Maximum value in upper triangular="+max);
    }

    public static void main(String s[ ]) throws Exception
    {
        atul ob=new atul( );
        ob. ass( );
    }
}

```

**P72) W.A.P. to find minimum value in upper triangular.**

```

import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int arr[][]=new int[3][3];

        int min;

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        System.out.print("Enter values in an array=");
    }
}

```



```

for(int i=0;i<3;i++)
{
    for(int j=0;j<3;j++)
    {
        arr[i][j]=Integer.parseInt(br.readLine());
    }
}
min=arr[0][1];
for(int i=0;i<3;i++)
{
    for(int j=0;j<3;j++)
    {
        if(i<j)
        {
            if(min>arr[i][j])
            {
                min=arr[i][j];
            }
        }
    }
}

System.out.print("Minimum value in an upper triangular="+min);
}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob.ass( );
}
}

```

**P73) W.A.P. to add all elements in lower triangular.**

```
import java.io.*;

class atul
{
    void ass( ) throws Exception
    {
        int arr[][]=new int[3][3];
        int s=0;
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter values in an array=");
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                arr[i][j]=Integer.parseInt(br.readLine());
            }
        }
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                if(i>j)
                {
                    s=s+arr[i][j];
                }
            }
        }
        System.out.print("Sum of all lower triangular elements=%d",s);
    }

    public static void main(String s[ ]) throws Exception
```

```

    {
        atul ob=new atul( );
        ob. ass( );
    }
}

```

**P74) W.A.P. to add all upper triangular elements.**

```

import java.io.*;
class atul
{
    void ass( ) throws Exception
    {
        int arr[][]=new int[3][3];
        int s=0;
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter values in an array=");
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                arr[i][j]=Integer.parseInt(br.readLine());
            }
        }
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                if(i<j)
                {
                    s=s+arr[i][j];
                }
            }
        }
    }
}

```

```

        }
    }
    System.out.print("Add all upper triangular elements="+s);
}
public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

**P75) W.A.P. to add all diagonal elements.**

```

import java.io.*;
class atul
{
    void ass( ) throws Exception
    {
        int arr[][]=new int[3][3];
        int s=0;
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter values in an array=");
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                arr[i][j]=Integer.parseInt(br.readLine());
            }
        }
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)

```

```

        {
            if(i==j)
            {
                s=s+arr[i][j];
            }
        }
    }

    System.out.print("Add all diagonal elements="+s);
}

public static void main(String s[ ]) throws Exception
{
    atul ob=new atul( );
    ob. ass( );
}
}

```

## Method in Java

There are two type of method used in java.

1)non-parameterize

2) Parameterize

1)non-parameterize

Ex:- class atul

```

{
    void add()
    {
        Int a=10,b=20,c;
        c=a+b;
        System.out.print("Add="+c);
    }
}

```

```

    }
    public static void main(String s[])
    {
        atul ob=new atul();
        ob.add();
    }
}

```

## 2) Parameterize:-

Ex:- class atul

```

{
    void add(int a,int b)
    {
        int c;
        c=a+b;
        System.out.print("Add="+c);
    }
    public static void main(String s[])
    {
        atul ob=new atul();
        ob.add(10,20);
    }
}

```

## Constructor in Java

In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling constructor, memory for the object is allocated in the memory.

## Rules for Creating a Constructor

- 1) Class name and constructor name must be same.
- 2) Constructor don't have any return type.
- 3) Constructor can't be static,final,abstract.
- 4) Constructor don't call constructor call it self.

## Types of Constructor

- 1) Default constructor (no-arg constructor)
- 2) Parameterized constructor

### 1) Default constructor (no-arg constructor):-

```
class atul
{
    atul()
    {
        System.out.print("Hello I am Atul");
    }

    public static void main(String a[])
    {
        atul ob=new atul();
    }
}
```

### 2) Parameterized constructor

```
class atul
{
```

```

atul(int a,int b)
{
    int c;
    c=a+b;
    System.out.print("Add="+c);
}

public static void main(String s[])
{
    atul ob=new atul(10,20);
}
}

```

## Static

### Static Variable

If you declare any variable as static, it is known as a static variable.

The static variable gets memory only once in the class area at the time of class loading.

Ex:-

```

class atul
{
    int a=0;
    atul()
    {
        a++; //incrementing value
        System.out.println(a);
    }
    public static void main(String args[])
    {
        atul c1=new atul();
        atul c2=new atul();
        atul c3=new atul();
    }
}

```



```
}
```

If we can't use variable as static then it is can't common for all objects. It will get a memory every time.

Ex:-

```
class atul
{
    static int a=0;
    atul()
    {
        a++; //incrementing value
        System.out.println(a);
    }
    public static void main(String args[])
    {
        atul c1=new atul();
        atul c2=new atul();
        atul c3=new atul();
    }
}
```

If we can use variable as static then it is common for all objects. It will get a memory one time.

## Static Method

- A static method belongs to the class rather than the object of a class.
- A static method can be invoked without the need for creating an instance of a class.
- A static method can access static data member and can change the value of it.

Ex:-

```
class atul
{
    static int a=10;
    static void as()
    {
        System.out.print(a);
    }
}
```

```

public static void main(String s[])
{
    atul ob=new atul();
    ob.as();
}
}

```

Static method only access the static data member. If we can access the non static data member in static method then it can show an error.

### Static Block

\*After loading the class the compiler executes the static block and then main function.

Ex:-

```

class atul
{
    static
    {
        System.out.print("Hello");
    }
}

```

If you are using JDK 1.7 OR MORE. Then it can show an error.

JDK 1.7 after loading the class then it can find static block and then main function.

### this keyword

1. this can be used to refer current class instance variable.
2. this can be used to invoke current class method (implicitly)
3. this() can be used to invoke current class constructor.
4. this can be passed as an argument in the method call.

5. this can be passed as argument in the constructor call.
6. this can be used to return the current class instance from the method.

Ex:-

```
class atul
{
    int a;
    atul(int a)
    {
        a=a;
    }
    void display()
    {
        System.out.println(a);
    }
    public static void main(String args[])
    {
        Student s1=new Student(10);
        Student s2=new Student(20);
        s1.display();
        s2.display();
    }
}
```

it can display 0 because it can't consider it is an local variable or Global.

Ex:-

```
class atul
{
    int a;
    atul(int a)
    {
        this.a=a;
    }
    void display()
    {
```

```

        System.out.println(a);
    }
    public static void main(String args[])
    {
        Student s1=new Student(10);
        Student s2=new Student(20);
        s1.display();
        s2.display();
    }
}

```

It display 10 20 because this can be used to refer current class instance variable.

## Inheritance

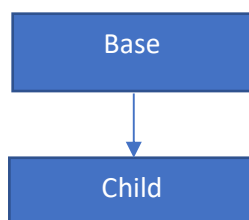
Inheritance is use to inherit the properties of base class in to child Class.

Type of Inheritance

- 1)single
- 2)Multi Level
- 3) Hierarchical
- 4)Multiple
- 5)Hybrid

1)single :-

A single base class inherit by the single child class .

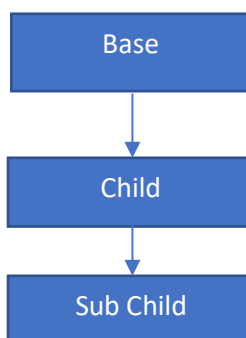


Ex:-

class base

```
{  
    void as()  
    {  
        System.out.print("Hello");  
    }  
}  
class child extends base  
{  
    void ps()  
    {  
        System.out.print("Hi");  
    }  
    public static void main(String s[])  
    {  
        child c=new child();  
        c.as();  
        c.ps();  
    }  
}
```

## 2)Multi Level



Ex:-

```
class base
{
    void as()
    {
        System.out.print("Hello");
    }
}

class child extends base
{
    void ps()
    {
        System.out.print("Hi");
    }
}

class Subchild extends child
{
    void ds()
    {
        System.out.print("Hi");
    }

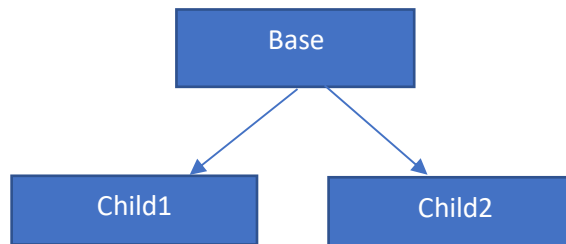
    public static void main(String s[])
    {
        Subchild ob=new Subchild();
        ob.as();
    }
}
```

```

        ob.ps();
        ob.ds();
    }
}

```

### 3) Hierarchical



```

class base
{
    void as()
    {
        System.out.print("Hello");
    }
}

class child1 extends base
{
    void ps()
    {
        System.out.print("HI");
    }
}

class child2 extends child1
{

```

```

void ds()
{
    System.out.print("HI");
}

public static void main(String s[])
{
    child1 ob=new child1();
    ob.as();
    ob.ps();
    child2 ob=new child2();
    obj.as();
    obj.ds();
}
}

```

#### 4)Multiple 5)Hybrid

These two inheritance we can't access because two class can't inherit at a time .

### Method Overloading

If a class has multiple methods having same name but different in parameters, it is known as **Method Overloading**.

**Ex:-**

```

class atul
{
    void add()

```



```

{
    int a=10,b=20,c;

    c=a+b;

    System.out.print("Add="+c);
}

void add(int p,int q)
{
    int r=p+q;

    System.out.print("Addition="+r);
}

public static void main(String s[])
{
    atul ob=new atul();

    ob.add();

    ob.add(10,20);
}
}

```

## Method Overriding

- Method overriding is used to provide the specific implementation of a method which is already provided by its superclass.
- Method overriding is used for runtime polymorphism
- The method must have the same name as in the parent class
- The method must have the same parameter as in the parent class.

Ex:-

```
class atul
{
    void as()
    {
        System.out.print("Hello");
    }
}

class atul1 extends atul
{
    void as()
    {
        System.out.print("Hi");
    }

    Public static void main(String s[])
    {
        atul1 ob =new atul1();
        ob.as();
    }
}
```

### Super keyword

The super keyword can also be used to invoke the parent class constructor.

```

class base
{
    base()
    {
        System.out.println("Hello");
    }
}
class child extends base
{
    child()
    {
        super();
        System.out.println("hi");
    }
    public static void main(String args[])
    {
        child ob=new child();
    }
}

```

### Final Keyword

1. variable
2. method
3. class

#### 1)final variable :-

Final variable value can't be change. It value is final.

```

class atul

```

```

{
    final a=10;

    void as()
    {
        a++;          //it shown an error final value can't be change

        System.out.print(a);
    }
}

```

```
    }  
  
    public static void main(String s[])  
  
    {  
  
        atul ob=new atul();  
  
        ob.as();  
  
    }  
  
}
```

## **2)final Mehtod:-**

Final method can't be override.

```
class atul  
  
{  
  
    final void as()  
  
    {  
  
        System.out.print("Hello");  
  
    }  
  
}  
  
class atul1 extends atul  
  
{  
  
    void as()  
  
    {  
  
        System.out.print("Hi");  
  
    }  
  
}
```

```
public static void main(String s[])
{
    atul1 ob=new atul1();
    ob.as();
}
}
```

### **3)final Class:-**

Final class can't be inherited.

final class base

```
{
    void as()
    {
        System.out.print("Hello");
    }
}
```

class child extends base

```
{
    void ps()
    {
        System.out.print("Hi");
    }

    public static void main(String s[])
```

```

    {
        Child ob=new child();
        ob.ps();
    }
}

```

## Upcasting

If we can create the reference variable of base class hold by the child class is called Upcasting.

class base

```

{
    void as()
    {
        System.out.print("Hello");
    }
}

```

Class child extends base

```

{
    void ps()
    {
        System.out.print("Hi");
    }
    public static void main(String s[])
    {
        base ob=new child();
    }
}

```

```

        ob.ps();
        ob.as();
    }
}

```

### Dynamic binding

Wrapping with data member and member function with in a single class is called dynamic binding.

```

class atul
{
    int a=10;
    void as()
    {
        System.out.print("Hello");
    }
    public static void main(String s[])
    {
        atul ob=new atul();
        ob.as();
    }
}

```

### Abstract Class

- An abstract class must be declared with an abstract keyword.
- It can have abstract and non-abstract methods.
- It cannot be instantiated.
- It can have constructors and static methods also.
- It can have final methods which will force the subclass not to change the body of the method.

```
abstract class atul
{
    abstract void as();
    void ps()
    {
        System.out.print("Hello");
    }
}
class atul1 extends atul
{
    void as()
    {
        System.out.print("Hi");
    }
    public static void main(String s[])
    {
        atul1 ob=new atul1();
        ob.as();
        ob.ps();
    }
}
```