CORE JAVA

Q. What is JAVA?

Java is pure object oriented language

Q. What is object oriented?

Object oriented is a concept of programming or it is standard methodology in programming.

Note: Java is not object oriented.
Java is programming language which implement Object oriented programming concept so java is object oriented.

If we want to work with java we should have to know
How to create application using JAVA

If we want to create application using java we have some important points or steps.
1. Download and Installed JDK
Q. What is JDK?

JDK stands for java development kit
Def: JDK is software cluster or group of software which contains some supporting software's those provide environment to us to create and develop the java application.
Like as it contain Compiler, JVM etc

Q. What is compiler?

Compiler is application or software which is used for convert your source code in to byte code.

Q. What is byte code?

Byte code is machine understandable code means byte code is not machine code it is intermediate format which easily convert in machine code with the help of JVM

Q. What is JVM?
JVM stands for Java Virtual Machine basically it is software or application present in JDK setup or JRE setup which is used for convert your byte code in to machine code as well as creating objects, perform garbage collection as well as manage threads etc
Note: we will JVM in depth at the time of JVM Architecture.

If we want to download the JDK you visit the following URL https://www.oracle.com/in/java/technologies/downloads/#jdk22-windows

2. Create Sample application

If we want to create any program in java we have some standard or generalize code format
Note: if we want to write code in java we have n number editors and IDE
Editors for java code: notepad, wordpad , notepad++,vscode etc
IDE: eclipse,spring tool suite, IntelliJ etc
Syntax:

```
access specifier  class classname
{     public static void main(String x[])
        {  write here your logics
        }
}
```

Example:

```
public class FirstSep
{
  public static void main(String x[])
  {
          System.out.println("good morning");
  }
}
```

public class FirstSep : here public is access specifier
Q. what is access specifier?

Access specifier is the some keywords which help us to apply restrictions on class and its member.
There are four types of access specifier?

public: public is access specifier which allow us we can access class member outside of class as wel as outside of package also

private : private means member cannot access outside of class within same package or outside package also.

protected: protected access specifier means member can access within child class of same package as well as outside of package.

default: it is known as package level access specifier means member can access outside of class but within same package.

class: class is keyword for class declaration purpose and FirstSep is class name and user can give any name to his class.

public static void main(String x[]): it is main function of java class same like as main function in c or c++

System.out.println("good morning"): it is output statement of java same like as printf() in c language.

Meaning: System is class out is static reference of PrintStream class so System and PrintStream maintain HAS-A relationship between and println() is overloaded method which help us to display the output on output screen.
Note: we will discuss this meaning in inheritance chapter.

3. Save Application: if we want to save code in java then save in bin folder where JDK install and give class name and file name same with .java extension.

4 Compile Applications

Q. What is compilation?
Compilation is process where we convert source code to byte code in java

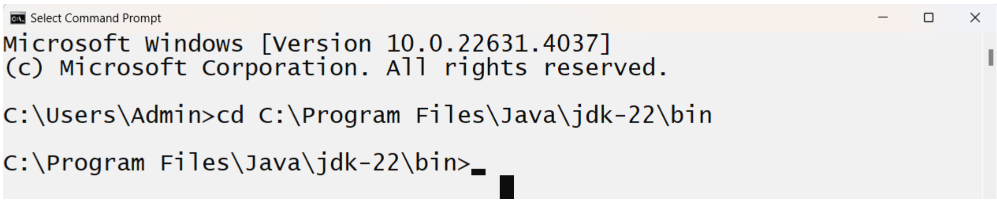
Q. Why java develop the byte code?

Byte code is platform independent code
Q. what is platform?
Platform means operating system means byte code can execute on any operating system without support of any third party application called as byte code

How to compile java code?

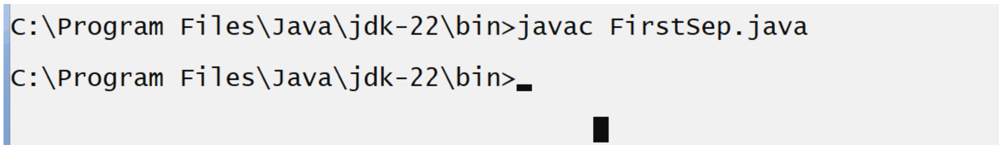
If we want to compile java code we have use following steps.

Step1:
A. open command prompt
start menu ----- search --- command prompt
B. go where java file save



```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd C:\Program Files\Java\jdk-22\bin

C:\Program Files\Java\jdk-22\bin>
```

C. type the command javac filename.java
Example: javac FirstSep.java



```
C:\Program Files\Java\jdk-22\bin>javac FirstSep.java

C:\Program Files\Java\jdk-22\bin>
```

If we think about above screen short out get compile successfully
Note: when your code compile successfully then java compiler create new file with extension of .class and in this file contain your byte code.

Means as per our example after compilation we have two files
FirstSep.java ---- source code
FirstSep.class --- byte code

5. Run Application: if you want to run your java application we have to use following command
java filename
java – it work as JVM internally i.e java.exe

Data Types in JAVA

____

Q. What is data type?

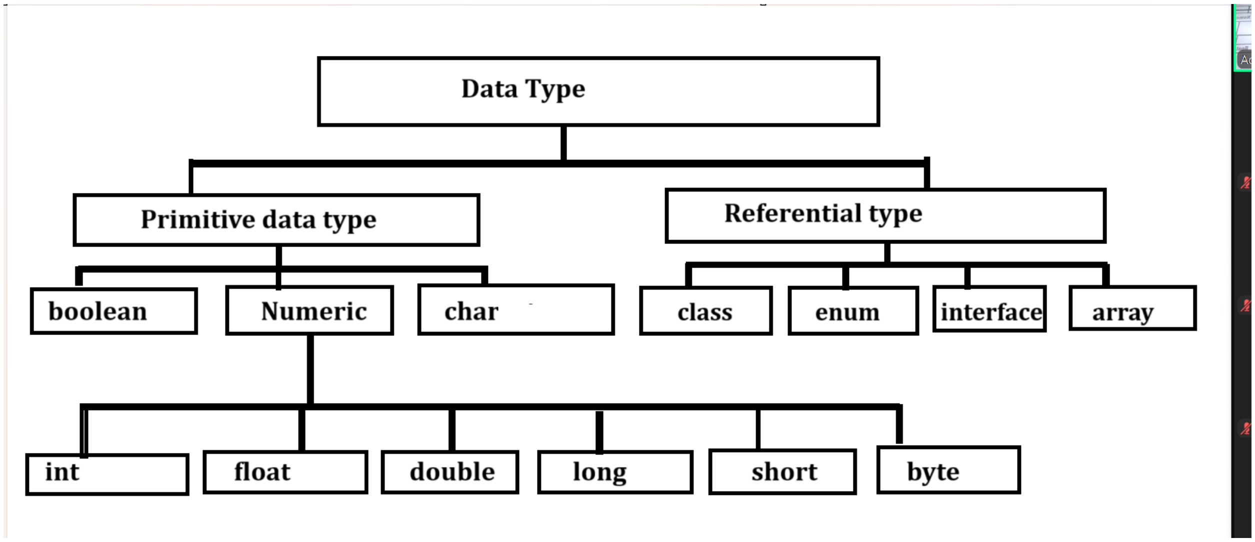
____

Data type means to decide what kind of information we want to use in program called as data type.

There are two types of data type

1. Primitive data type: primitive data type means those data type already provided by java to us and not able to store address of another memory called as primitive type of data.
Example: int , float , double ,long etc

2. Referential data type / Non-Primitive: referential data type means those data type able to store address of another memory called as referential data type and referential data type contain may be user defined data type.



Now we want to discuss about primitive type of data

Integer data type

If we want to work with any data type we need to know the four important points.

How to use in code: write simply int

How much memory required: 4 byte

Q. Why 4 byte?

It is not fixed it is dependent on compiler

Q. Can we see the memory size of data type?

Yes we can see the memory size of data type in JAVA

Write code for check memory size of integer

```
public class CheckMemorySizeApp
{  public static void main(String x[])
   {          int size=Integer.SIZE;
              System.out.println((size/8));
   }
}
```

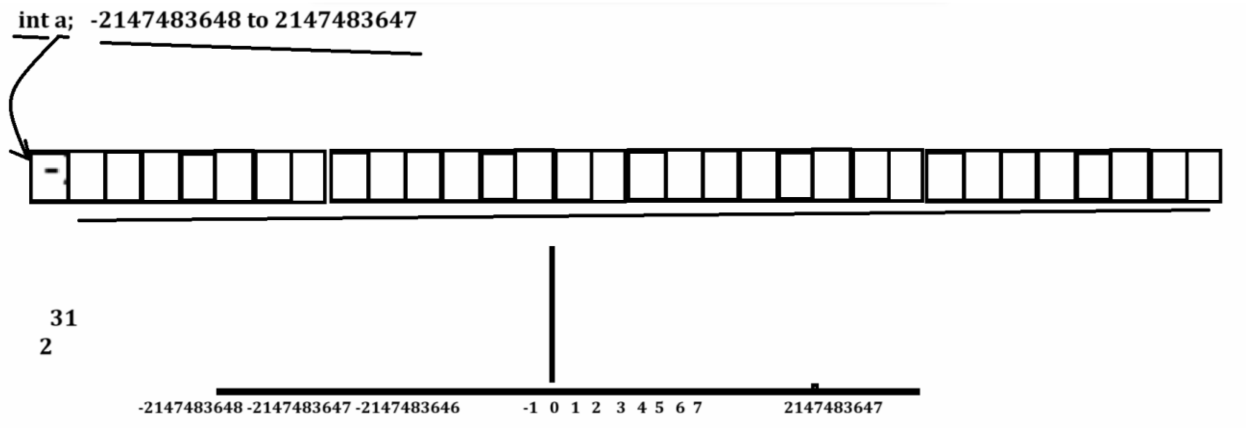
Range of data type: range decide the value storage capacity of data type or Calculation capacity of data type.

If we think about integer data type we have range of integer data type

-2147483648 to 2147483647

How we can calculate this range?

Range of data type is dependent on memory size of data type.

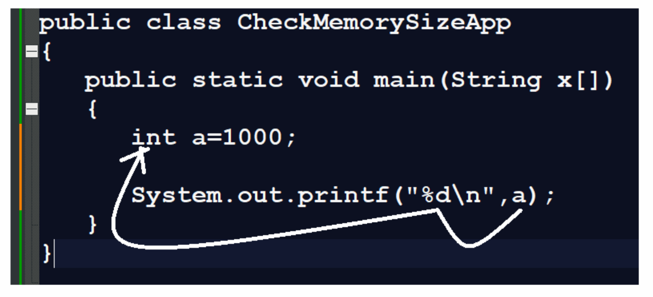


Note: if we think about above diagram we can say integer required 4 byte memory means 32 bit of memory so 1 bit required sign or (negative sign) and remaining 31 bits required for value so the logic is

31
2

Format specifier of data type: %d

Note: format specifier data type of variable in printf()



```
public class CheckMemorySizeApp
{
   public static void main(String x[])
   {
      int a=1000;

      System.out.printf("%d\n",a);
   }
}
```

Now we want to discuss about the long data type

How to use in code: long

Memory size: 8 byte or x 64 bit

Range:

63
2

Format specifier: %d

Short integer

___

How to use in code: short int

Memory required: 2 byte

Range: -32768 to 32767

15
2

byte data

___

How to use in code: byte

Memory required : 1 byte

Range: -128 to 127

or

7
2

Format specifier: %d

Character data type

___

Character type specially designs for hold the alpha numeric values.

How to use in code: char
Memory required: 2 byte
Range: **-**32768 to 32767 or 2^{15}
Format specifier: %c

Boolean

___

This data type is used for store true or false value

How to use in code: boolean
Memory required: 1 byte
Format specifier: %b

float data type

___

If we think about float data type it is used for store floating values or decimal point values.

There are two types of float data type

1. float

how to use in code: float
memory required : 4 byte
Range: -3.14×10^{38} to 3.14×10^{38}
Format specifier : %f

2. double

How to use in code: double
Memory required: 8 byte
Range: -1.7×10^{308} to 1.7×10^{308}
Format specifier: %f

Operator in JAVA

___

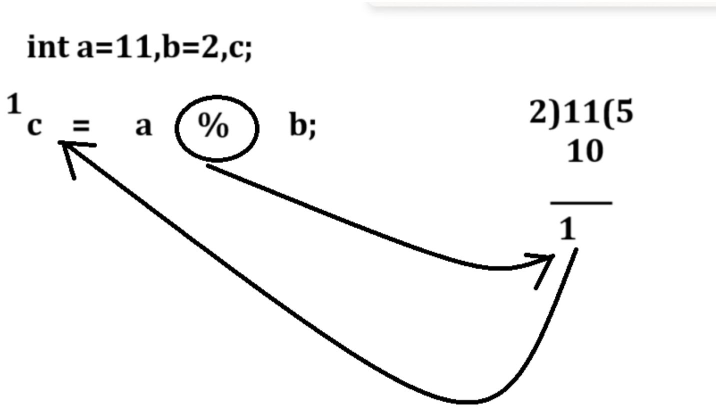
Operators the some symbol which are used for perform some operations.

There are seven types of operator in JAVA

1. Arithmetic operator: Arithmetic operator is used for perform arithmetic operations to us

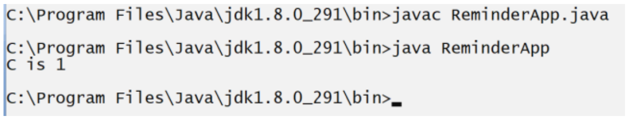
Operator	Meaning
+	Addition
-	Substraction
*	Multiplication
/	Division
%	Reminder or modules operator

Example of reminder operator or modules operator



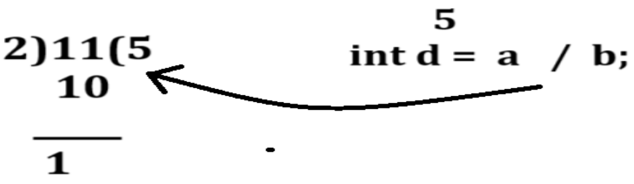
Example with source code

```
public class ReminderApp
{
  public static void main(String x[])
  {
    int a=11,b=2,c;
           c=a%b;
           System.out.printf("C is %d\n",c);
  }
}
```



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ReminderApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java ReminderApp
C is 1

C:\Program Files\Java\jdk1.8.0_291\bin>_
```

Example of division operator

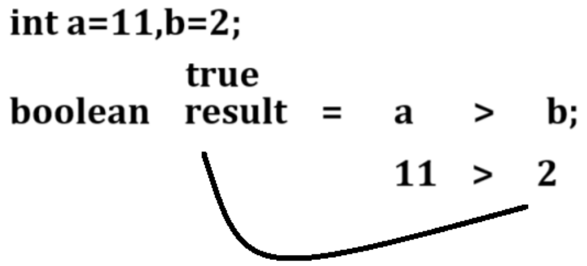


2. Assignment operator: Assignment operator is used for assign value from right hand side to left hand side variable.

Operator	**Meaning**
=	**Assignment**

3. Relational operator: Relational operator is used for perform comparison and if comparison get success return true otherwise return false

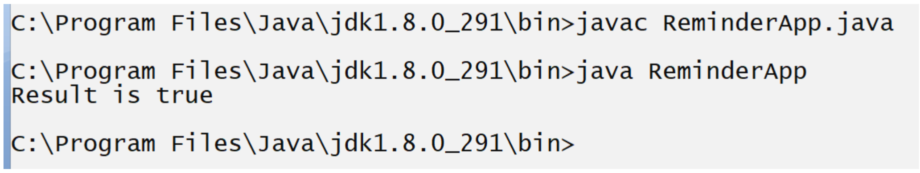
Operator	**Meaning**
<	Less than
>	Greater than
<=	Less than equal
>=	Greater than equal
==	Equals
!=	Not equals



Example with source code

```
public class ReminderApp
{
  public static void main(String x[])
  {
    int a=11,b=2;
           boolean c=a>b;
           System.out.printf("Result is %b\n",c);
  }
}
```

Output

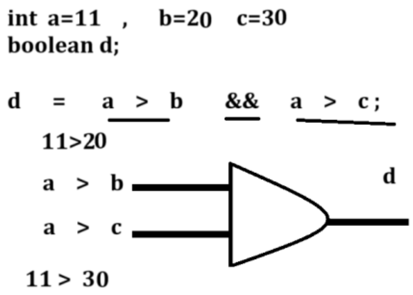


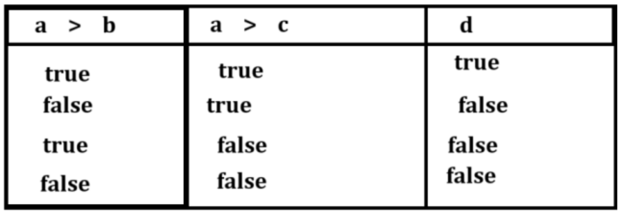
```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ReminderApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java ReminderApp
Result is true

C:\Program Files\Java\jdk1.8.0_291\bin>
```

4. Logical Operator: Logical operator is used for combine more than one condition and mark as single condition and check it called as logical operator

Operator	Meaning
&&	Logical AND – if all conditions are true then condition is true otherwise condition is false.
||	Logical OR – if any condition is true then condition is true otherwise condition is false.
!	Logical NOT – if condition is true then false and if false then true

Note: Every logical operator work as logical gate internally.



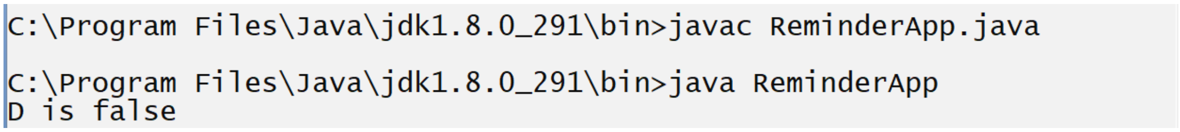


a > b	a > c	d
true	true	true
false	true	false
true	false	false
false	false	false

Example with source code

```
public class ReminderApp
{
  public static void main(String x[])
  {
    int a,b,c;
           boolean d;
           a=11;
           b=20;
           c=30;
           d=a>b && a>c;
           System.out.printf("D is %b\n",d);
  }
}
```

Output



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ReminderApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java ReminderApp
D is false
```

5. Increment and Decrement Operator

Increment and decrement operator is used for increase value by 1 and decrease value by 1

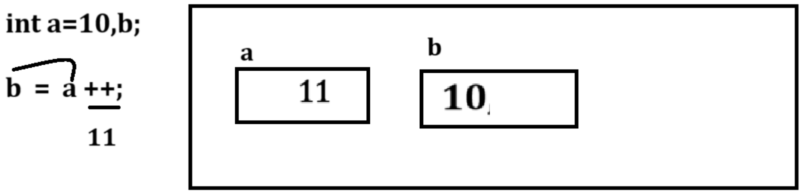
Operator	Meaning
++	**Increment by 1**
--	**Decrement by 1**

Note: if we think about increment and decrement operator we have two types of increment and decrement operator.

a) Post increment and post decrement: post increment and post decrement means first we shift value at left hand side and after that we can perform increment or decrement operation on it.

Syntax: variablename op;

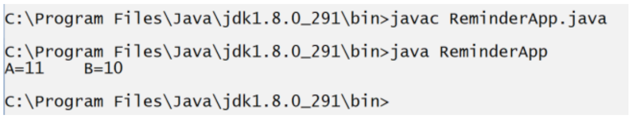
Example of post increment



Example with source code

```
public class ReminderApp
{
  public static void main(String x[])
  {
    int a=10,b;
           b=a++;
           System.out.printf("A=%d\tB=%d\n",a,b);
  }
}
```

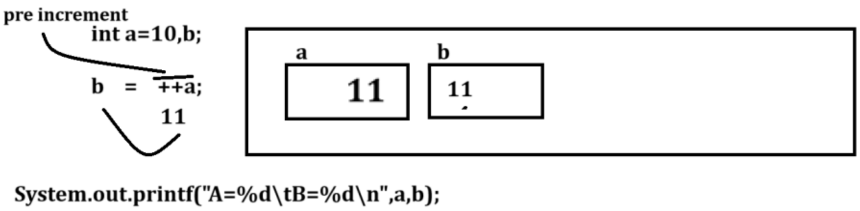
Output



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ReminderApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java ReminderApp
A=11    B=10

C:\Program Files\Java\jdk1.8.0_291\bin>
```

b) Pre increment and pre decrement: first perform operation and after that value shift at left hand side

Syntax: op variable name;



Example with source code

```
public class ReminderApp
{   public static void main(String x[])
  {      int a=10,b;
             b=++a;
             System.out.printf("A=%d\tB=%d\n",a,b);
  }
}
```

Output

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ReminderApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java ReminderApp
A=11    B=11

C:\Program Files\Java\jdk1.8.0_291\bin>
```

6. Conditional Operator: conditional operator is used for check the conditions

Operator	**Meaning**
?	**Ternary operator** :Ternary operator those operator help us to combine three expression
:	**Colon operator or option operator**

If we want to work with conditional operator we have some standard format.

Syntax: data type variable name   =   exp1   ?   exp2   :   exp3;

exp1: is always condition if exp1 is true then exp2 get executed if exp1 is false then exp3 get executed.

exp2: means logic which we want to execute when exp1 is true

exp3: means logic which we want to execute then exp1 is false.

data type variable name: this variable help us to hold result send by exp2 or exp3 according to exp1 result.

Example with source code

```
public class ReminderApp
{   public static void main(String x[])
  {  int a,b;   String s;
             a=11;
             b=2;
             s = a > b ? "A is Greater" : "B is Greater";
             System.out.println(s);
  }
}
```

Output

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ReminderApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java ReminderApp
A is Greater
```

If we think about above code we fix the values a=11 and b=2 means we can compare only 11 and 2 with each other so if we want to perform comparison with another values then we need to modify the values manually in code and it is not possible every time so we want to accept the input from keyboard after program run and then perform comparison

How to accept the input from keyboard after program run using JAVA

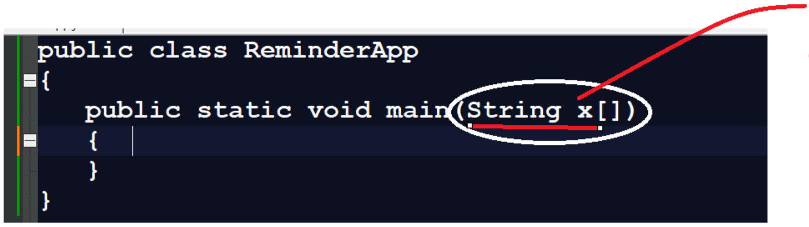
You can accept from keyboard in java by using two ways

a) Using Command Line Argument
b) Using Scanner class

Now we want to discuss about Command Line argument

Q. What is command line argument?

Command line argument is parameter present in main function of String array called as command line argument shown in following diagram



```
public class ReminderApp
{
   public static void main(String x[])
   {
   }
}
```

It is command line argument

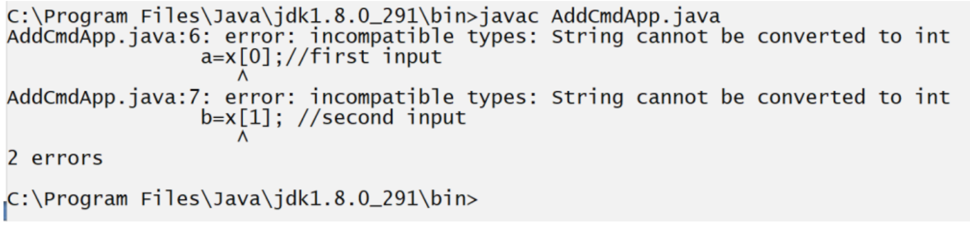
Note: Basically command line argument is infinite string array present in main function and which is used for accept input from keyboard in the form of string means using command line argument we can accept infinite number of inputs but the first input is at position of zero to n-1.

Example: WAP to input two values and calculate its addition using command line arguments

---

```
public class AddCmdApp
{
  public static void main(String x[])
        {
           int a,b,c;
                 a=x[0];//first input
                 b=x[1]; //second input
                 c=a+b;
                 System.out.printf("Addition is %d\n",c);
        }
}
```

Output



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac AddCmdApp.java
AddCmdApp.java:6: error: incompatible types: String cannot be converted to int
                a=x[0];//first input
                   ^
AddCmdApp.java:7: error: incompatible types: String cannot be converted to int
                b=x[1]; //second input
                   ^
2 errors

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Note: if we think about above code we get compile time error String cannot be converted to int

Q. Why?

Because we have string for input and we accept input of type integer and string cannot store directly in integer so compiler will generate error to us incompatible types.

Means if we think about above code we have statement int a=x[0]; here x[0] is type of string and a is type of integer means we try to store string type of value in to integer type so compiler will generate compile time error to us.

How to solve this type of error in JAVA?

If we want to solve this type of error in java we have use type casting technique.

Q. What is type casting?

Type casting means convert one type of data in to another type for single line of code called as type casting.

Means as per our example we required to convert our string value in to integer value.

How to Convert String value to integer value in JAVA

if we want to convert string value to integer value in java we have following statement

Syntax: int variable =Integer.parseInt(String): Here Integer is class and parseInt() function of Integer class which help us to convert string value to integer value.

Note: if we think about Java if we want to identify class then we have some standard because in java class name start with capital letter and if class name contain more than one word then every word first letter should be capital.

Example: **Integer:** here Integer is single word class so I should capital.

DataInputStream: here this class form by using three different word Data , Input , Stream so D , I and S must be capital.

Example of class name : **EmployeeSalaryManagment**

Note: if we think about function in java then if first form by using single word then all letters of word must be small but if function name form by more than words then first word all letters must be small and remaining all word first letter must be capital.

Example: read(): all letter small because read() is single word function

Example: readLine(): here first word all letter small but second word L must be capital because Line() is second word not first

Example: getEmployeeSalary():

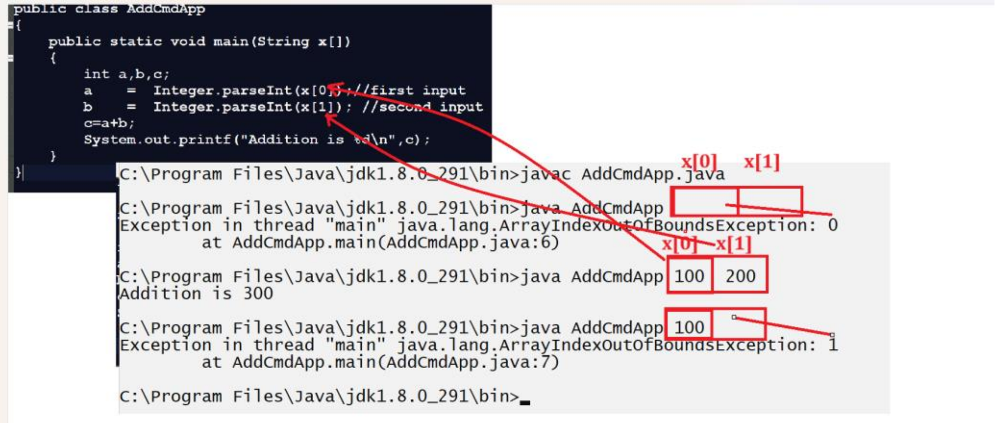
findEmployeeByUsingAge()

How to convert string to float value

Syntax: float variable =Float.parseFloat(String): convert string to float data type

How to convert String to double value

Syntax: double variable=Double.parseDouble(String): convert string to double type value.

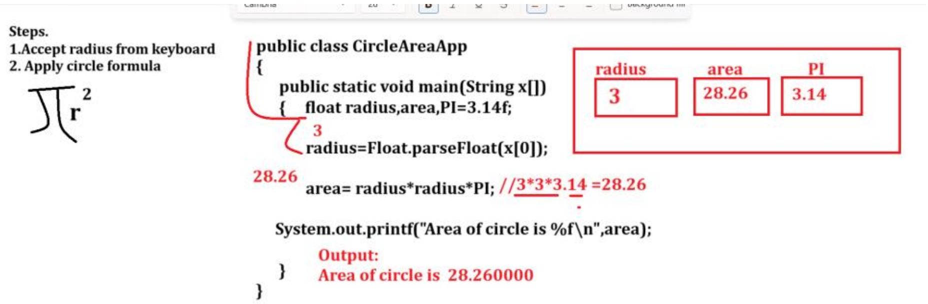


```
public class AddCmdApp
{
    public static void main(String x[])
    {
        int a,b,c;
        a    =  Integer.parseInt(x[0]);//first input
        b    =  Integer.parseInt(x[1]); //second input
        c=a+b;
        System.out.printf("Addition is %d\n",c);
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac AddCmdApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java AddCmdApp
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0
        at AddCmdApp.main(AddCmdApp.java:6)

C:\Program Files\Java\jdk1.8.0_291\bin>java AddCmdApp 100 200
Addition is 300

C:\Program Files\Java\jdk1.8.0_291\bin>java AddCmdApp 100
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 1
        at AddCmdApp.main(AddCmdApp.java:7)

C:\Program Files\Java\jdk1.8.0_291\bin>_
```

Note: if we use command line argument then all input must be provide on same line where we run your java program if we give less input and press enter then we get index of that input as run time error.

Example: WAP to input radius of circle and calculate its area



Steps.
1.Accept radius from keyboard
2. Apply circle formula

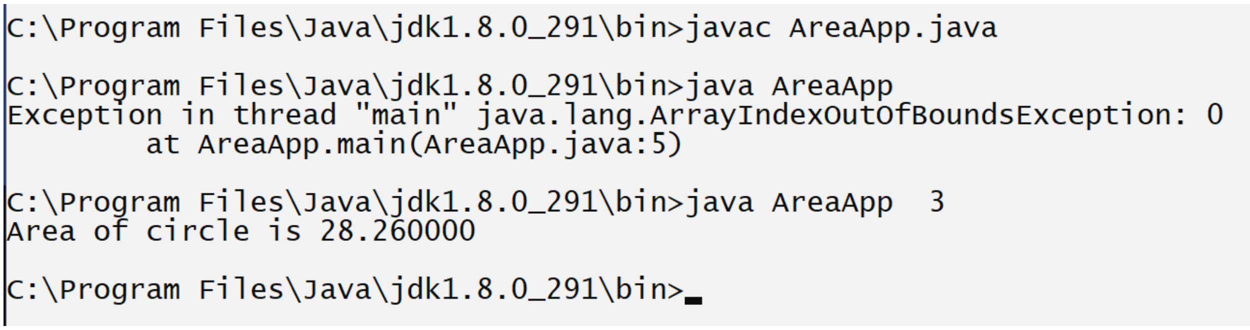
πr^2

```
public class CircleAreaApp
{
   public static void main(String x[])
   {   float radius,area,PI=3.14f;

       radius=Float.parseFloat(x[0]);

       area= radius*radius*PI; //3*3*3.14 =28.26

   System.out.printf("Area of circle is %f\n",area);

   }
}
```

Output:
Area of circle is 28.260000

Example with source code and output

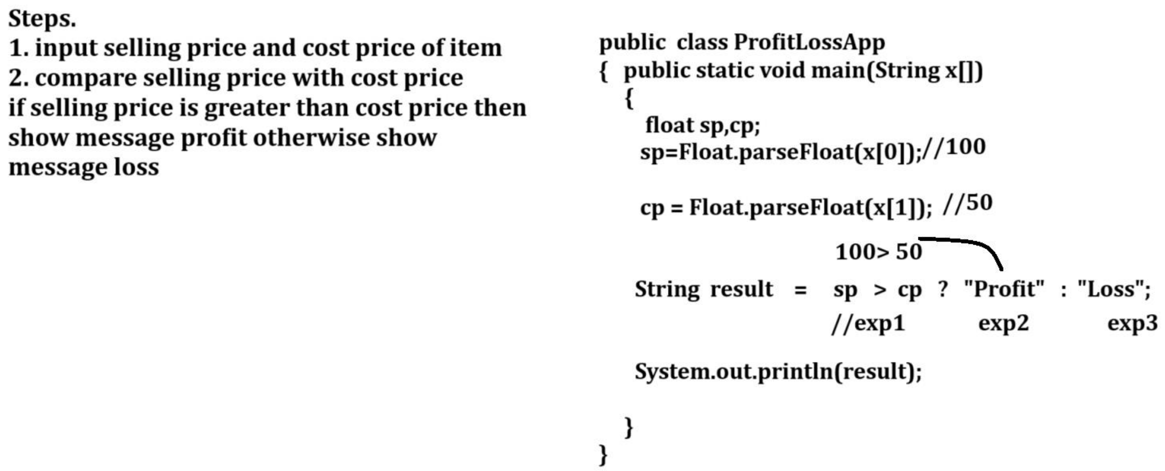
```
public class AreaApp
{
  public static void main(String x[])
  {  float radius,area,PI=3.14f;
    radius=Float.parseFloat(x[0]);
           area=radius*radius*PI;
           System.out.printf("Area of circle is %f\n",area);
  }
}
```



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac AreaApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java AreaApp
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0
        at AreaApp.main(AreaApp.java:5)

C:\Program Files\Java\jdk1.8.0_291\bin>java AreaApp  3
Area of circle is 28.260000

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Example: WAP input the selling price of item and cost price item from keyboard and check seller made profit or loss using command line arguments.

Note: use conditional operators



Steps.
1. input selling price and cost price of item
2. compare selling price with cost price
if selling price is greater than cost price then show message profit otherwise show message loss

```
public  class ProfitLossApp
{  public static void main(String x[])
   {
     float sp,cp;
    sp=Float.parseFloat(x[0]);//100

    cp = Float.parseFloat(x[1]); //50
                       100> 50
   String  result   =  sp  > cp  ?  "Profit"  : "Loss";
                       //exp1       exp2        exp3

   System.out.println(result);

   }
}
```

Example with nested conditional operator

```
public class ProfitLossApp
{
  public static void main(String x[])
  {
                  float sp,cp;
                  sp=Float.parseFloat(x[0]);
```

```
                    cp=Float.parseFloat(x[1]);

                    String result   =   ((sp==cp) ?  "No Profit No Loss" :(sp>cp?"Profit":"Loss"));

                    System.out.println(result);
  }
}
```

If we think about command line arguments then we can accept input on same line we cannot accept input on new line so if we want to accept input on new line then we have to use Scanner class.

Scanner class

Steps to work with Scanner class

a) Add java.util package in application

Q. What is package?

Package is a collection of classes and interfaces it is like as header file in c
If we want to add package in program we have to use import keyword.
Syntax: import packagename.*: if we use * then we can import all classes and interfaces from package in application.

or

import packagename.classname: if we this syntax then we can import specific member from package in application.

Example: import java.util.*;
or
import java.util.Scanner;

b) Create object of Scanner class

Syntax: Scanner ref = new Scanner(System.in);
Example: Scanner xyz = new Scanner(System.in);

c) Use Scanner class method to accept input from keyboard

If we want to accept the input from keyboard Scanner class provide some inbuilt method to us
int nextInt(): this method can accept input of type integer
float nextFloat(): this method can accept input of type float
double nextDouble(): this method can accept input of type double
long nextLong(): this method can accept input of type long
short nextShort(): this method is used for accept the input of type short

String nextLine(): this method is used for accept the input of type string.

etc

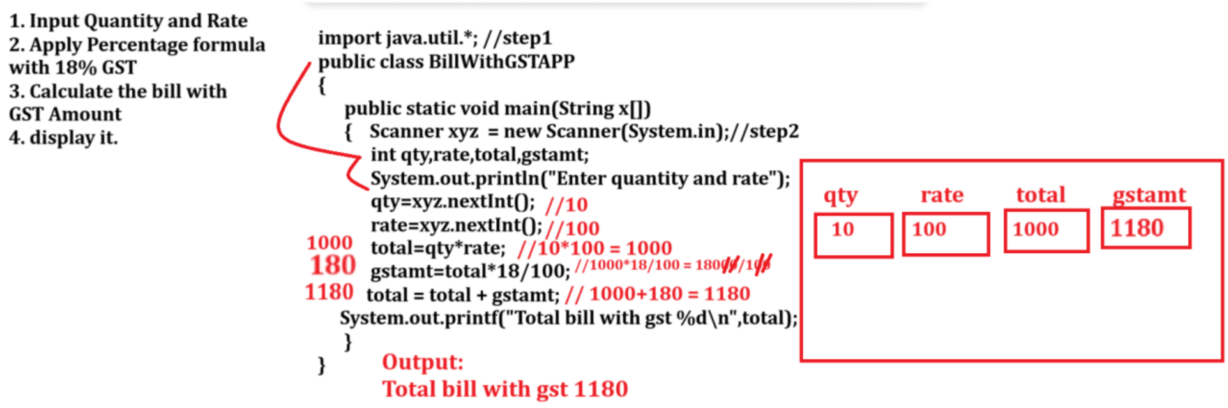
If we want to use method use the following syntax

Syntax: data type variablename = scannerref.methodname();

Example: we want to accept two values from keyboard and calculate its addition.

```
import java.util.*; //step1
public class AddScanApp
{     public static void main(String x[])
            { Scanner xyz  = new Scanner(System.in);//step2
              int a,b,c;
              System.out.println("Enter two values");
              a=xyz.nextInt(); //step3
              b=xyz.nextInt(); //step4
              c=a+b;
              System.out.printf("Addition is %d\n",c);
            }
}
```

Example: WAP to input the quantity and rate of product using Scanner class and apply 18% percentage of GST on bill and calculate total bill.



1. Input Quantity and Rate
2. Apply Percentage formula with 18% GST
3. Calculate the bill with GST Amount
4. display it.

```
import java.util.*; //step1
public class BillWithGSTAPP
{
    public static void main(String x[])
    {   Scanner xyz  = new Scanner(System.in);//step2
        int qty,rate,total,gstamt;
        System.out.println("Enter quantity and rate");
        qty=xyz.nextInt();  //10
        rate=xyz.nextInt(); //100
1000    total=qty*rate;  //10*100 = 1000
180     gstamt=total*18/100; //1000*18/100 = 180~~00~~/1~~00~~
1180    total = total + gstamt; // 1000+180 = 1180
      System.out.printf("Total bill with gst %d\n",total);
    }
}
```

Output:
Total bill with gst 1180

qty	rate	total	gstamt
10	100	1000	1180

Example with source code

```
import java.util.*;//step1
public class BillWithGSTAPP
{
    public static void main(String x[])
          {
            int qty,rate,total,gstAmt;
                   Scanner xyz = new Scanner(System.in);
                   System.out.println("Enter quantity and rate");
                   qty=xyz.nextInt();
```

```
                        rate=xyz.nextInt();
                        total=qty*rate;
                        System.out.printf("\nBill Without GST %d\n",total);
                        gstAmt=total*18/100;
                        total=total+gstAmt;
                        System.out.printf("Total bill with gst %d\n",total);
            }
}
```

Example: WAP to input the basic salary of employee and calculate its da and hra using following terms and calculate total bill

da=30%

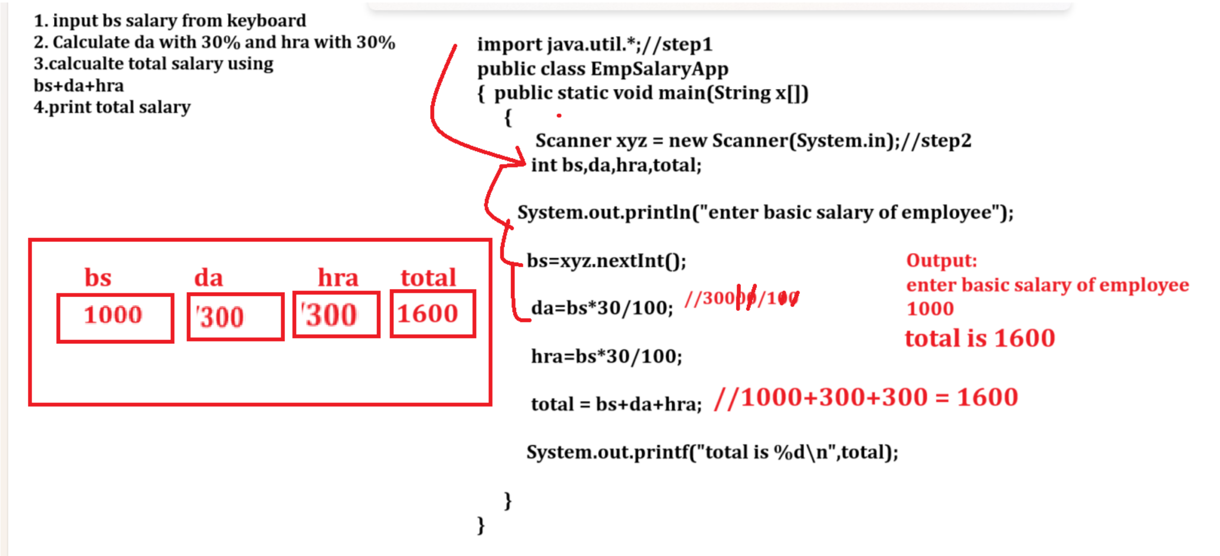
hra=30%

bs=1000

da=300

hra=300

total=bs+da+hra ;// 1600



1. input bs salary from keyboard
2. Calculate da with 30% and hra with 30%
3.calcualte total salary using
bs+da+hra
4.print total salary

```
import java.util.*;//step1
public class EmpSalaryApp
{ public static void main(String x[])
   {
     Scanner xyz = new Scanner(System.in);//step2
    int bs,da,hra,total;

   System.out.println("enter basic salary of employee");

    bs=xyz.nextInt();

    da=bs*30/100;  //300~~00~~/1~~00~~

    hra=bs*30/100;

    total = bs+da+hra;  //1000+300+300 = 1600

    System.out.printf("total is %d\n",total);

   }
}
```

Output:
enter basic salary of employee
1000
total is 1600

Example with source code

```
import java.util.*;//step1;
public class EmployeeGrossSalaryApp
{
    public static void main(String x[])
            {  Scanner xyz  = new Scanner(System.in);
                        int bs,da,hra,total;
                        System.out.println("Enter basic salary of employee");
                        bs=xyz.nextInt();
                        da=bs*30/100;
```

```
                    hra=bs*30/100;
                    total=bs+da+hra;
                    System.out.printf("total salary of employee is %d\n",total);
          }
}
```

Example: WAP to input principle amount , rate of interest and duration of loan amount and calculate the monthly emi of loan amount.

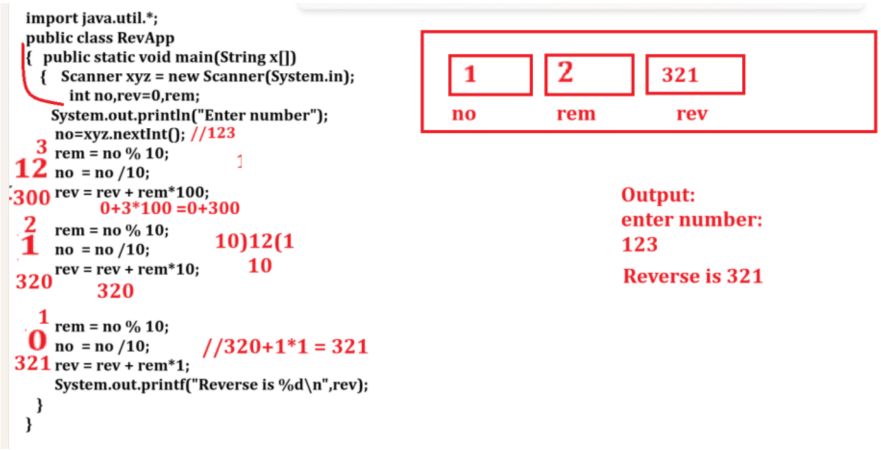
Example with source code

```
import java.util.*; //step1
public class SimpleInterestApp
{       public static void main(String x[])
          { Scanner xyz  = new Scanner(System.in);//step2
           float pAmt,irate,dur,emi;
           System.out.println("Enter principle amount");
            pAmt=xyz.nextFloat();
            irate=xyz.nextFloat();
           dur=xyz.nextFloat();
            float o=pAmt/dur;
            emi=pAmt*irate*dur/100;
            System.out.printf("Emi is %f\n",(emi+o)/12);
          }
}
```

Example: WAP to input the three digit number and reverse it.

Input: 123

Output: 321



```
import java.util.*;
public class RevApp
{  public static void main(String x[])
   {   Scanner xyz = new Scanner(System.in);
       int no,rev=0,rem;
     System.out.println("Enter number");
      no=xyz.nextInt(); //123
   3  rem = no % 10;
  12  no  = no /10;
 300  rev = rev + rem*100;
             0+3*100 =0+300
   2  rem = no % 10;
   1  no  = no /10;                10)12(1
      rev = rev + rem*10;              10
 320         320
   1  rem = no % 10;
   0  no  = no /10;          //320+1*1 = 321
 321  rev = rev + rem*1;
      System.out.printf("Reverse is %d\n",rev);
   }
}
```

```
Output:
enter number:
123
Reverse is 321
```

Example with source code

import java.util.*;

```
public class ThreeDigitRevApp
{
   public static void main(String x[])
          {
            Scanner xyz  = new Scanner(System.in);
            int no,rem,rev=0;
            System.out.println("Enter three digit number");
            no=xyz.nextInt();
            System.out.printf("Before Reverse is %d\n",no);
            rem=no%10;
            no=no/10;
            rev=rev+rem*100;

            rem=no%10;
            no=no/10;
            rev=rev+rem*10;

            rem=no%10;
            no=no/10;
            rev=rev+rem*1;
             System.out.printf("After reverse is %d\n",rev);

          }
}
```

no=123;

$\boxed{321}$

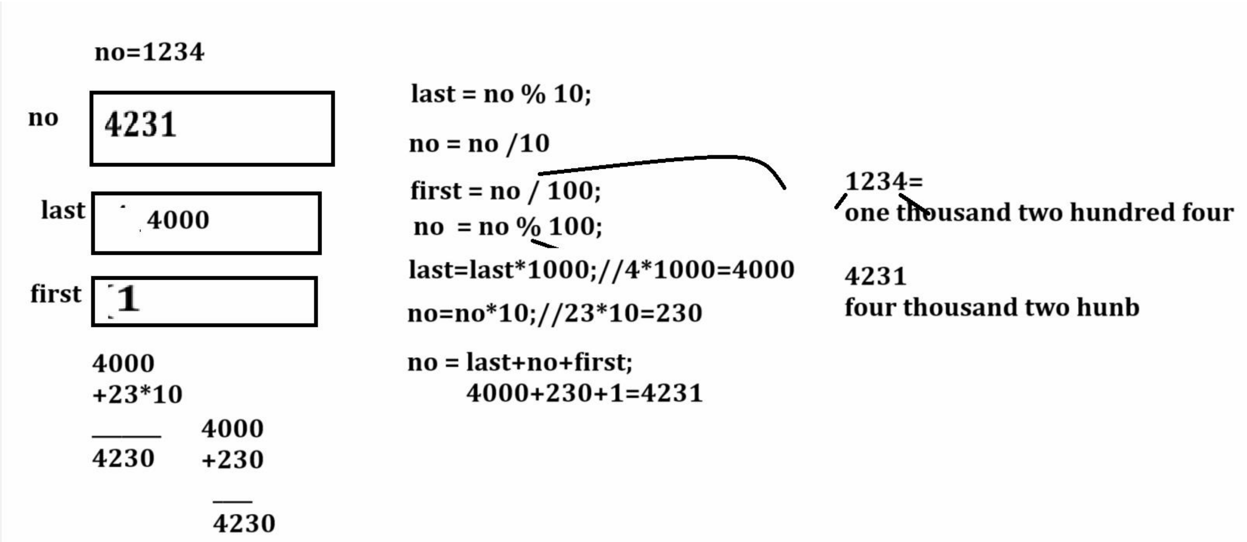
```
                   3                      12%10    2           1
     no  =  ( no % 10) * 100 + ( ( no/10 ) %10) * 10 + no /100;
321
            10)123(12               10)123(12              100)123(1
                 120                     120
                 ___                     ___
                   3                       3

            3*100 +2*10+ 1
            300+20+1 = 321
```

or

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

```
            {
              Scanner xyz  = new Scanner(System.in);
              int no,rem,rev=0;
              System.out.println("Enter three digit number");
              no=xyz.nextInt();
              System.out.printf("Before Reverse is %d\n",no);
              no=(no%10)*100 +((no/10)%10)*10 + no/100;
              System.out.printf("After reverse is %d\n",no);
            }
}
```

Example: WAP to input four digit number and swap only first and last digit



no=1234

no 4231

last 4000

first 1

4000
+23*10
4230

4000
+230
4230

```
last = no % 10;
no = no /10
first = no / 100;
no  = no % 100;
last=last*1000;//4*1000=4000
no=no*10;//23*10=230
no = last+no+first;
     4000+230+1=4231
```

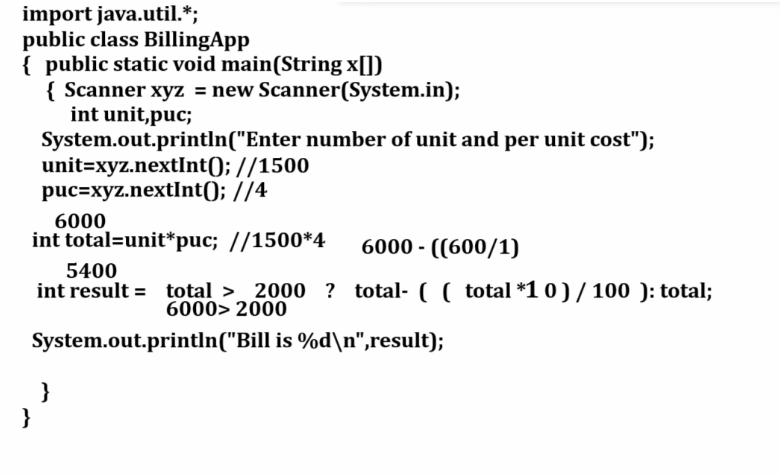
1234=
one thousand two hundred four

4231
four thousand two hunb

Example with source code

```
import java.util.*;
public class SwapFirstAndLastDigitApp
{
   public static void main(String x[])
            {  Scanner xyz  = new Scanner(System.in);
              int no,last,first;
              System.out.println("Enter four digit number");
              no=xyz.nextInt();
              System.out.printf("Before Swapping  %d\n",no);
              last=no%10;
              no=no/10;
              first=no/100;
              no=no%100;
              last=last*1000;
              no=no*10;
            no=last+no+first;
            System.out.printf("After Swapping %d\n",no);
        }
}
```

Example: WAP to input the electricity meter unit and per unit rate and calculate its total bill and If bill is greater than 2000 then give discount 10% otherwise there no discount



```
import java.util.*;
public class BillingApp
{  public static void main(String x[])
   { Scanner xyz  = new Scanner(System.in);
      int unit,puc;
  System.out.println("Enter number of unit and per unit cost");
  unit=xyz.nextInt(); //1500
  puc=xyz.nextInt(); //4
   6000
 int total=unit*puc;  //1500*4      6000 - ((600/1)
     5400
 int result =   total  >   2000   ?   total- (  (  total *1 0 ) / 100  ): total;
                6000> 2000
 System.out.println("Bill is %d\n",result);

  }
}
```

Example with source code

```
import java.util.*;
public class BillingApp
{      public static void main(String x[])
        {       int qty,rate,total;
                Scanner xyz  = new Scanner(System.in);
                System.out.println("Enter number of unit and per unit rate");
                qty=xyz.nextInt();
                rate=xyz.nextInt();
                total=qty*rate;
                int result=total>2000?total-((total*10)/100):total;
                System.out.printf("Total bill is %d\n",result);
        }
}
```

7. Bitwise Operator

Bitwise operator is used for perform operation on bits

If we want to work with bitwise operator properly we need to know the some important points.

1. Number System
2. Conversions of number system
3. One’s Complement and Two’s Complement
4. Binary operation on AND, OR and XOR

Number System

Q. What is Number System?

Number System means how we can present character input from keyboard in computer system called as Number system.

There are four types of number system

1. Binary Number System: Binary Number System it contain two digit i.e 0 & 1 and the base of binary number system is 2

Q. what is base?

Base means number of digit present in number system called as base and using base we can identify the number as per the computer system and base help in conversion of number system.

$$(111011)_{2} \text{ — base}$$

2. Octal Number System: If we think about octal number system it contains 0 to 7 numbers means total 8 number presents in octal so the base of octal number system is 8.

$$(12345)_{8}$$

3. Hexa Decimal Number System: it contain total 16 digits from 0 to 15 and after 10 there some character representation like as 10 =A, 11=B,12=C,13=D,14=E,15=F so the base of octal number system is 16.

(01234AF)
16

4. Decimal number system: decimal number system means it contain digit 0 to 9 so the total number of digit is 10 so the base of decimal is 10.

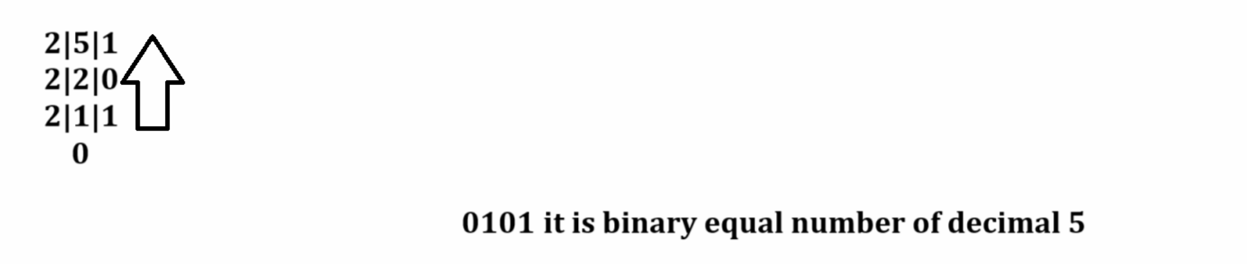
Example: (12345)
10

Conversion of Number System

Conversion helps to understand the number representation between human and computer.

1. Decimal to binary number system conversion

If we want to perform conversion between decimal to binary number then we need to divide decimal number by 2 and calculate its reminder and put reminders at right hand side and divide number until number is zero



2|5|1

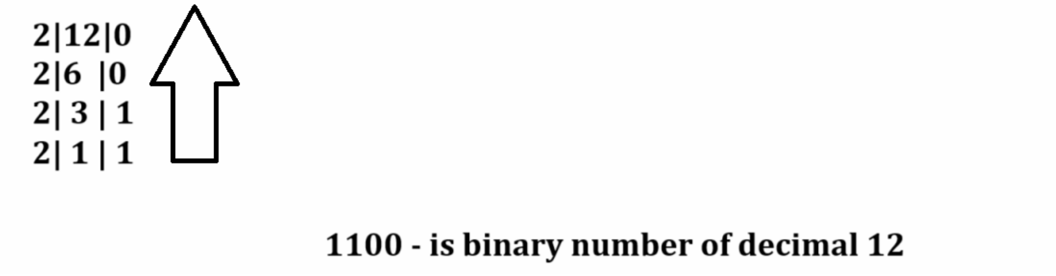
2|2|0

2|1|1

0

0101 it is binary equal number of decimal 5

Example: find the binary number of decimal 12?



2|12|0

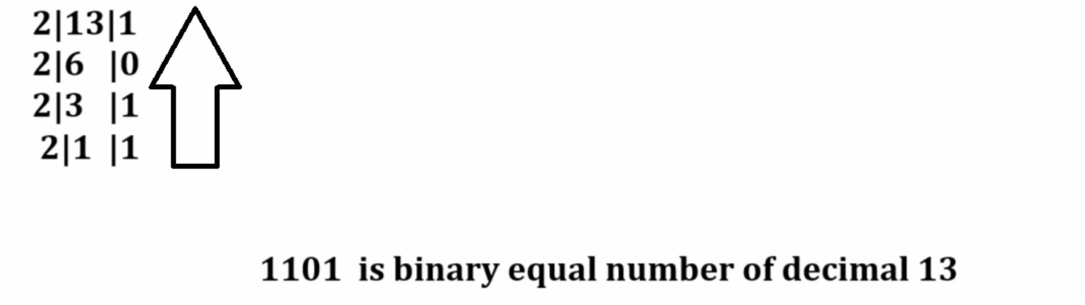
2|6 |0

2| 3 | 1

2| 1 | 1

1100 - is binary number of decimal 12

Example: Find the binary of decimal 13?



2|13|1

2|6 |0

2|3 |1

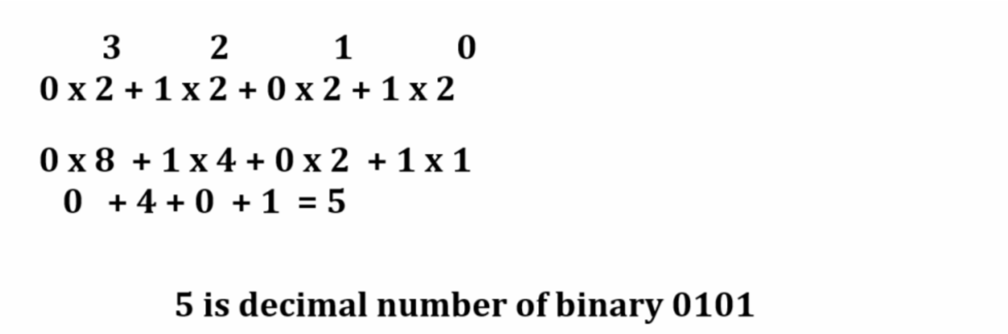
2|1 |1

1101 is binary equal number of decimal 13

How to convert binary to decimal number?

If we want to perform conversion between binary to decimal we required to multiply every binary digit by 2 and arrange its indexes from right to left and start index from 0 to n-1

Example: find the decimal number of 0101 binary?



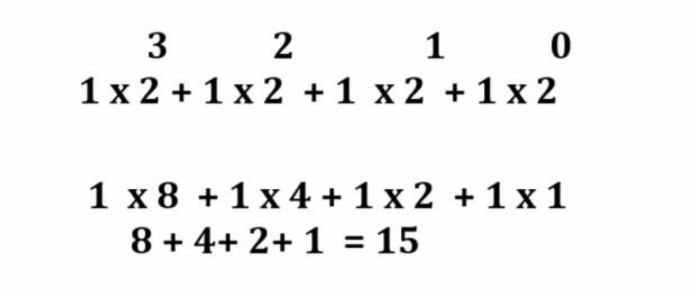
$0 \times 2^{3} + 1 \times 2^{2} + 0 \times 2^{1} + 1 \times 2^{0}$

$0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$

$0 + 4 + 0 + 1 = 5$

5 is decimal number of binary 0101

Example: Find the decimal of 1111?



$1 \times 2^{3} + 1 \times 2^{2} + 1 \times 2^{1} + 1 \times 2^{0}$

$1 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1$

$8 + 4 + 2 + 1 = 15$

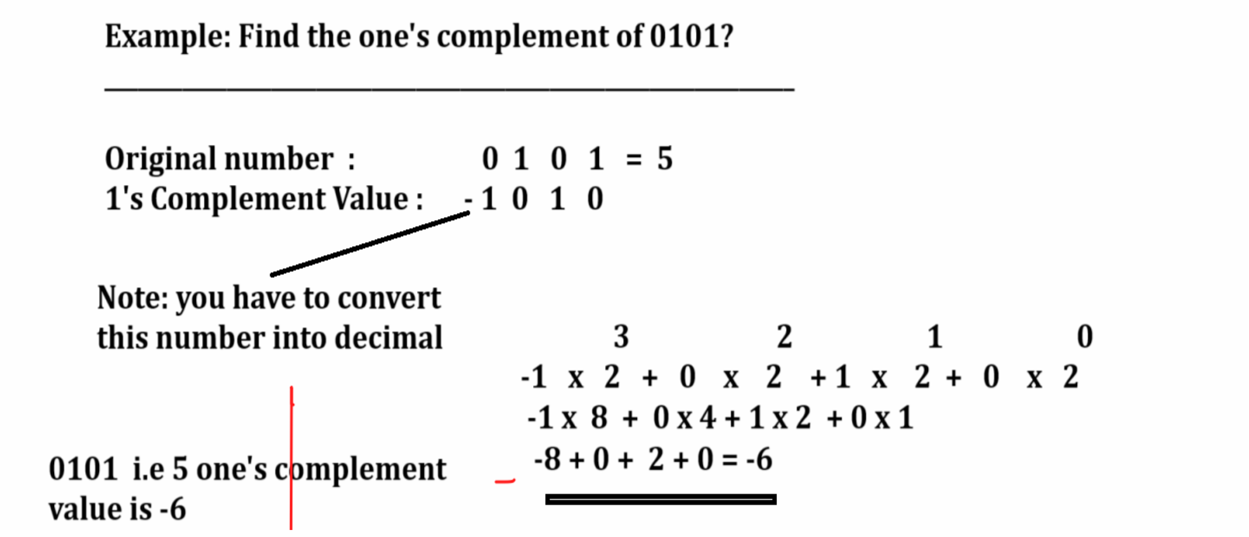
1’s complement and 2’s complement?

____

1’s complement?

____

1’s complement is negation operation on binary number means if we think about one’s complement then we convert 1 to 0 and 0 to 1



Example: Find the one's complement of 0101?

____

Original number : 0 1 0 1 = 5

1's Complement Value : - 1 0 1 0

Note: you have to convert this number into decimal

$-1 \times 2^{3} + 0 \times 2^{2} + 1 \times 2^{1} + 0 \times 2^{0}$

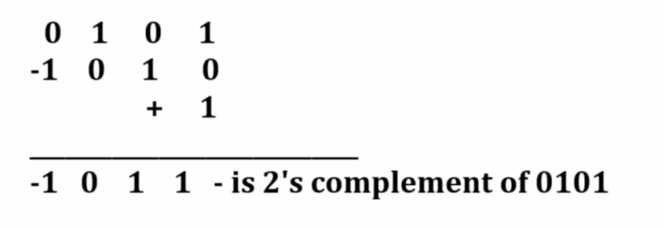
$-1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1$

$\underline{\underline{-8 + 0 + 2 + 0 = -6}}$

0101 i.e 5 one's complement value is -6

2’s complement

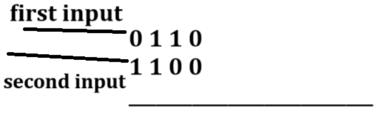
2’s complement means add 1 at right hand side in 1’s complement number and carry forward 1 if required

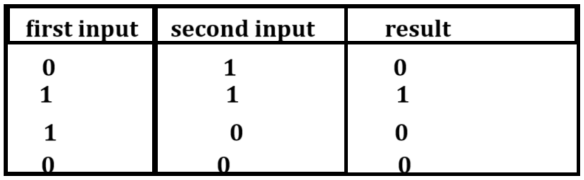


```
  0  1  0  1
 -1  0  1  0
        +  1
 ____________________
 -1  0  1  1  - is 2's complement of 0101
```

Binary AND, OR, XOR Operation on binary digit

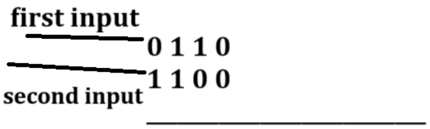
Binary AND: if we think about binary AND if all inputs are 1 then input is 1 otherwise 0.

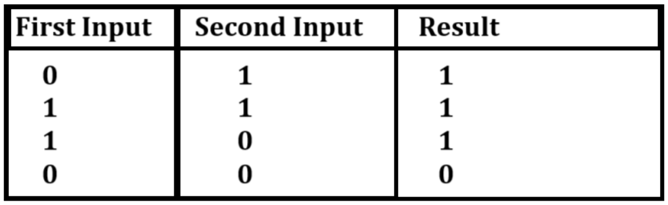




first input	second input	result
0	1	0
1	1	1
1	0	0
0	0	0

Binary OR Operation: Binary OR Operation means if any one input is 1 then output is 1 otherwise 0 means if all input 0 then output is 0.

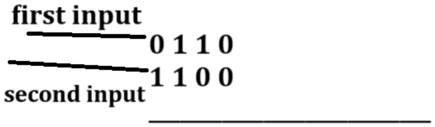


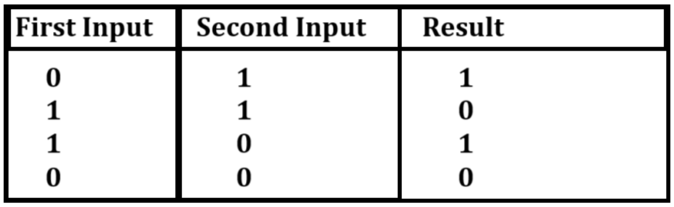


First Input	Second Input	Result
0	1	1
1	1	1
1	0	1
0	0	0

Binary XOR Operation

If we think about binary XOR Operation if we have same input then output is 0 and if we have different input then output is 1.



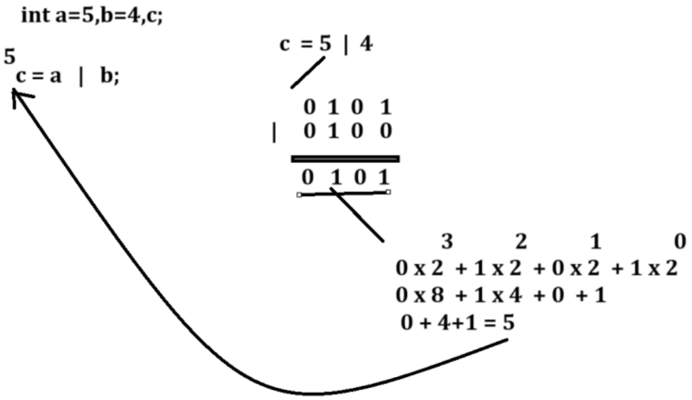


First Input	Second Input	Result
0	1	1
1	1	0
1	0	1
0	0	0

Bitwise Operator

Operator	Meaning
|	**Bit wise OR:** These operators convert given decimal value in to binary and performs OR Operation on it and generate result and convert in decimal and store at left hand side.
&	**Bitwise AND:** this operator convert given decimal value in to binary and perform AND Operation on it and generate result and convert in decimal and store at left hand side
<<	**Left shift operator:** this operator can shift bit from right hand side to left hand side and 0's at right hand side and convert result in to decimal and store at left hand side variable
>>	**Right Shift Operator:** this operator can shift bit from left to right and 0's at left hand side and generate result in binary and convert in decimal and stored left hand side
^	**XOR Operator:** this operator can convert decimal values in to binary and perform XOR operation on it and generate result and convert in decimal format at store at left hand side.
~	**One's Complement:** this operator can perform one's complement operation on given number

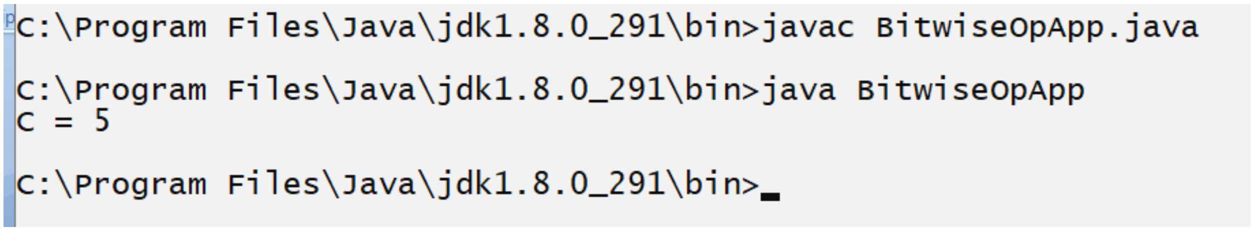
Now we want to discuss about bitwise OR Operator



Example with source code

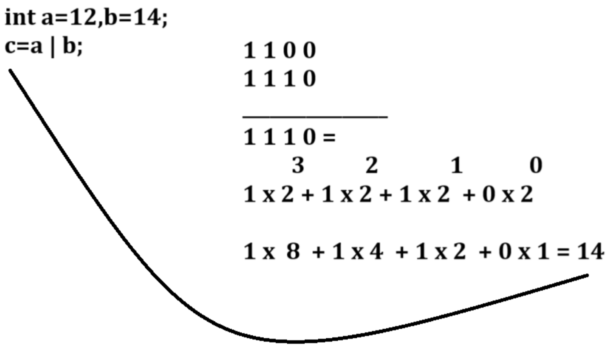
```
public class BitwiseOpApp
{
  public static void main(String x[])
  {
        int a,b,c;
        a=5;
        b=4;
        c=a|b;
        System.out.printf("C = %d\n",c);
  }
}
```

Output



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac BitwiseOpApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java BitwiseOpApp
C = 5

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Example:



int a=12,b=14;
c=a | b;

1 1 0 0
1 1 1 0

1 1 1 0 =

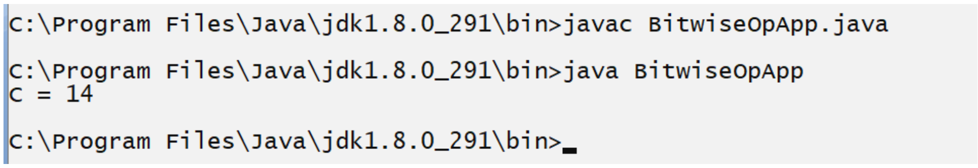
$1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

$1 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 14$

Example with source code

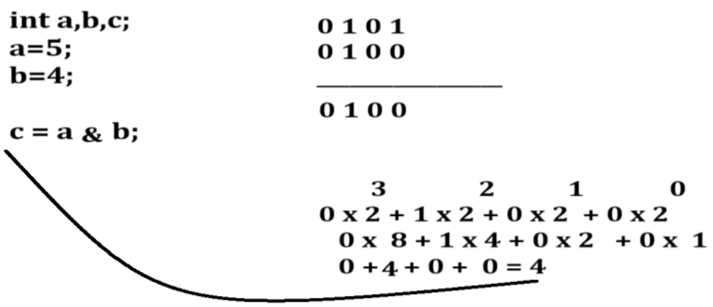
```
public class BitwiseOpApp
{   public static void main(String x[])
  {      int a,b,c;
        a=12;
        b=14;
        c=a|b;
        System.out.printf("C = %d\n",c);
  }
}
```

Output



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac BitwiseOpApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java BitwiseOpApp
C = 14

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Bitwise AND Operator



```
int a,b,c;
a=5;
b=4;

c = a & b;
```

0 1 0 1
0 1 0 0
____
0 1 0 0

$0 \times 2^{3} + 1 \times 2^{2} + 0 \times 2^{1} + 0 \times 2^{0}$

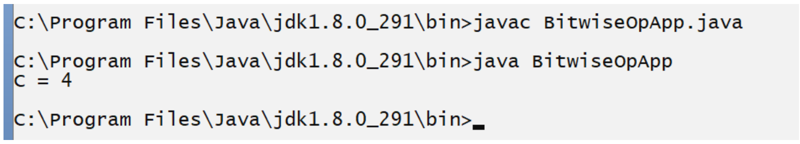
$0 \times 8 + 1 \times 4 + 0 \times 2 + 0 \times 1$

$0 + 4 + 0 + 0 = 4$

Example with source code

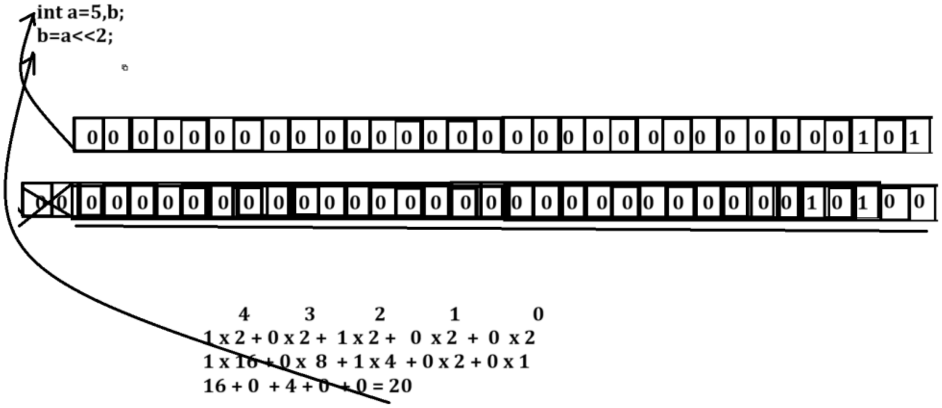
```
public class BitwiseOpApp
{
   public static void main(String x[])
   {
              int a,b,c;
              a=5;
              b=4;
              c=a&b;
              System.out.printf("C = %d\n",c);
   }
}
```

Output



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac BitwiseOpApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java BitwiseOpApp
C = 4

C:\Program Files\Java\jdk1.8.0_291\bin>_
```

Now we want to discuss about left shift operator



```
int a=5,b;
b=a<<2;
```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1

~~0 0~~ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0

$1 \times 2^{4} + 0 \times 2^{3} + 1 \times 2^{2} + 0 \times 2^{1} + 0 \times 2^{0}$

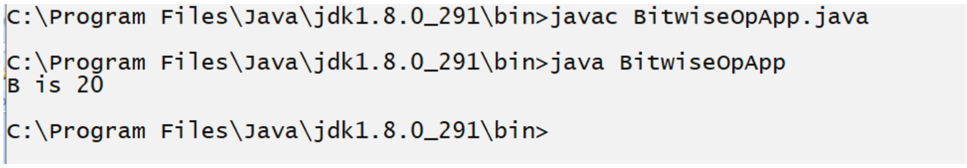
$1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 0 \times 1$

$16 + 0 + 4 + 0 + 0 = 20$

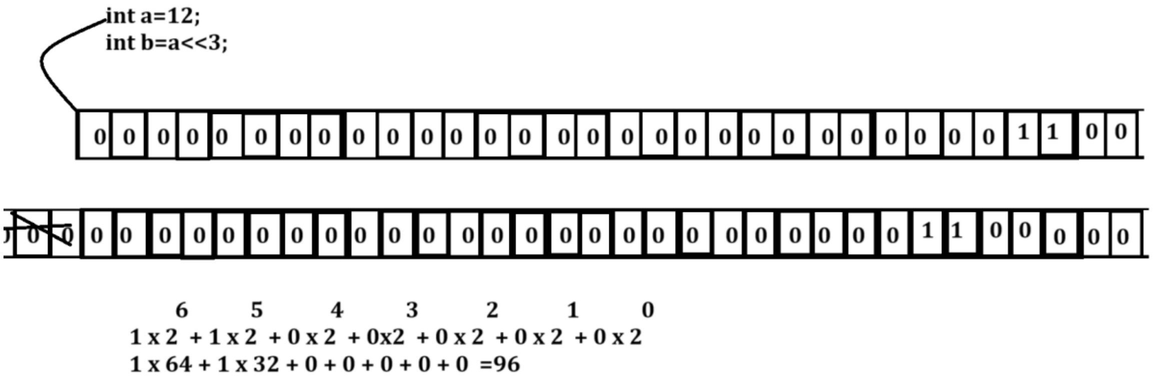
Example with source code

```
public class BitwiseOpApp
{
  public static void main(String x[])
  {
            int a,b,c;
            a=5;
            b=a<<2;
    System.out.println("B is "+b);
  }
}
```

Output



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac BitwiseOpApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java BitwiseOpApp
B is 20

C:\Program Files\Java\jdk1.8.0_291\bin>
```



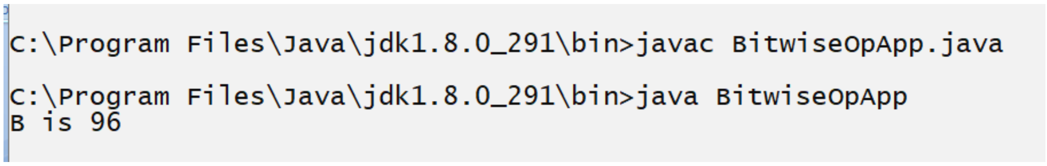
$1 x 2^{6} + 1 x 2^{5} + 0 x 2^{4} + 0x2^{3} + 0 x 2^{2} + 0 x 2^{1} + 0 x 2^{0}$

1 x 64 + 1 x 32 + 0 + 0 + 0 + 0 + 0 =96

Example with source code

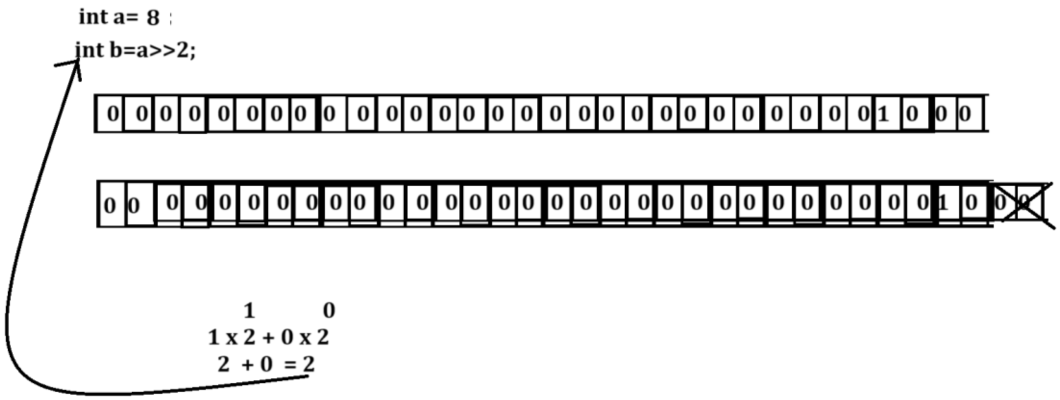
```
public class BitwiseOpApp
{
  public static void main(String x[])
    {int a,b,c;
            a=12;
            b=a<<3;
    System.out.println("B is "+b);
  }
}
```

Output



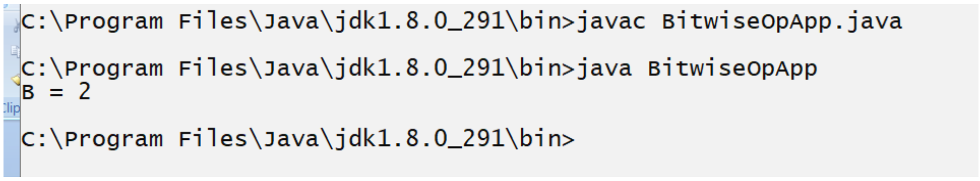
```
C:\Program Files\Java\jdk1.8.0_291\bin>javac BitwiseOpApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java BitwiseOpApp
B is 96
```

Right shift operator



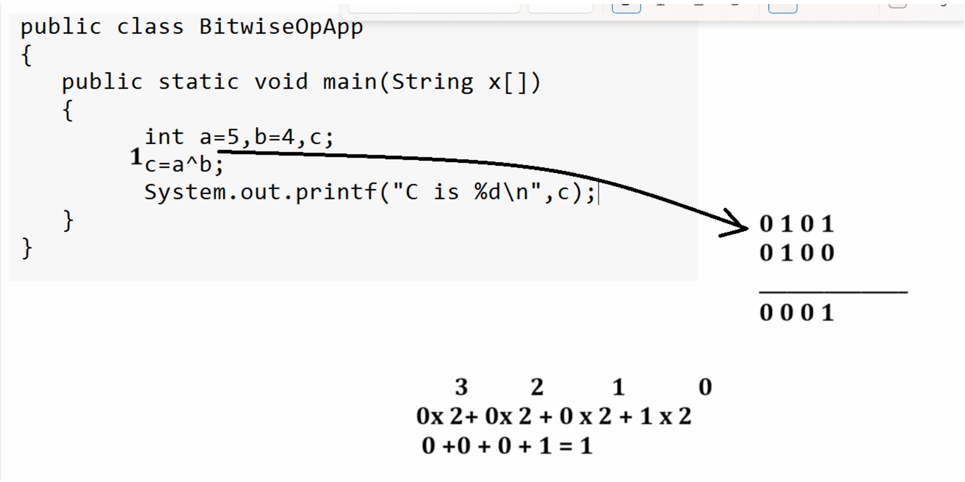
Example with source code

```
public class BitwiseOpApp
{
  public static void main(String x[])
  {
            int a=8,b;
      b=a>>2;
           System.out.printf("B = %d\n",b);
  }
}
```



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac BitwiseOpApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java BitwiseOpApp
B = 2

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Now we want to discuss about XOR Operator



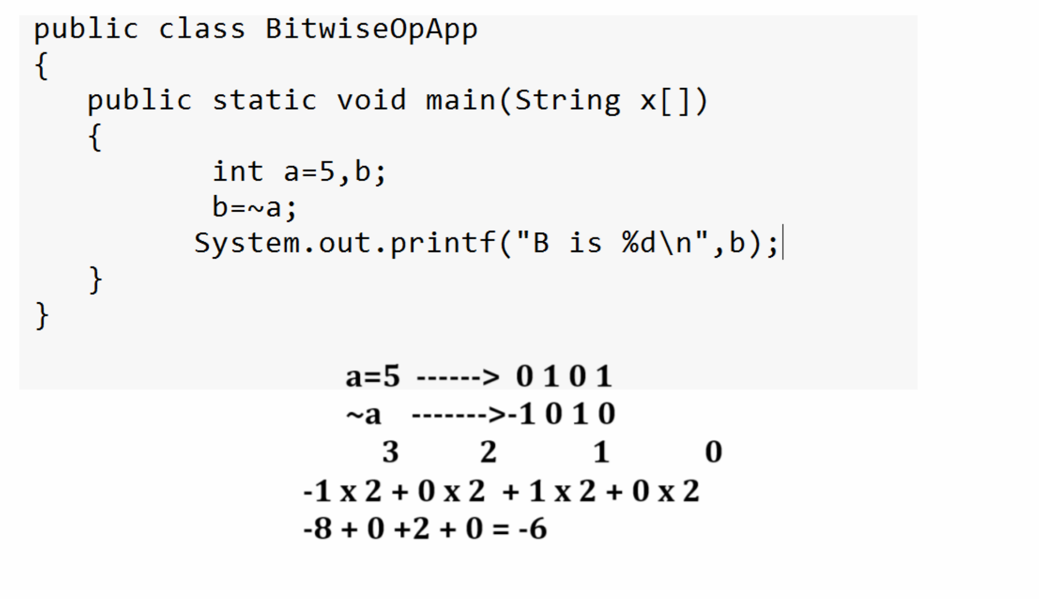
```
public class BitwiseOpApp
{
   public static void main(String x[])
   {
         int a=5,b=4,c;
        1c=a^b;
         System.out.printf("C is %d\n",c);
   }
}
```

```
0 1 0 1
0 1 0 0
_______
0 0 0 1
```

$0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$0 + 0 + 0 + 1 = 1$

Now we want to discuss about one's complement



```
public class BitwiseOpApp
{
   public static void main(String x[])
   {
          int a=5,b;
          b=~a;
         System.out.printf("B is %d\n",b);
   }
}
```

a=5 ------> 0 1 0 1

~a -------->-1 0 1 0

$$-1 \times 2^{3} + 0 \times 2^{2} + 1 \times 2^{1} + 0 \times 2^{0}$$

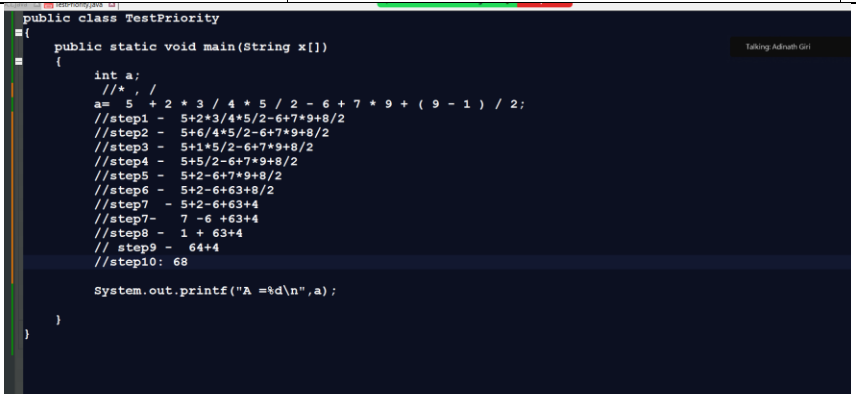
-8 + 0 +2 + 0 = -6

Operator Priority Concept

Operator priority means to decide execution priority of operator when multiple operator come in single expression called as operator priority.

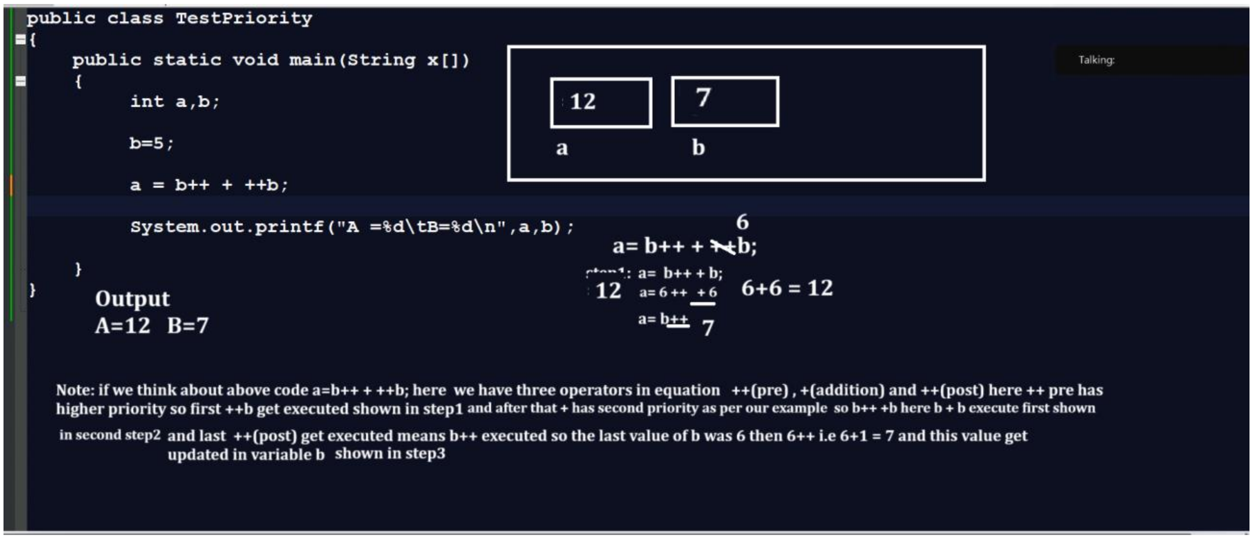
Precedence	Operator	Association
1	() []	Left to right
2	++ (pre) --	Right to left
3	~ (unary) - !	Right to left
4	* / %	Left to right
5	+ -	Left to right
6	<< >> >>>(bitwise operator with zero extension)	Left to right
7	<	Left to right

	<= > >= Instanceof	
8	== !=	Left to right
9	&	Left to right
10	^	Left to right
11	|	Bitwise OR
12	&&	Logical AND
13	||	Logical OR
14	? :	Right to left
15	++ -- (post)	Left to right
16	+=, -= , *= etc	Right to left



```
public class TestPriority
{
    public static void main(String x[])
    {
        int a;
         //* , /
        a=  5  + 2 * 3 / 4 * 5 / 2 - 6 + 7 * 9 + ( 9 - 1 ) / 2;
        //step1 -  5+2*3/4*5/2-6+7*9+8/2
        //step2 -  5+6/4*5/2-6+7*9+8/2
        //step3 -  5+1*5/2-6+7*9+8/2
        //step4 -  5+5/2-6+7*9+8/2
        //step5 -  5+2-6+7*9+8/2
        //step6 -  5+2-6+63+8/2
        //step7  - 5+2-6+63+4
        //step7-   7 -6 +63+4
        //step8 -  1 + 63+4
        // step9 -  64+4
        //step10: 68

        System.out.printf("A =%d\n",a);

    }
}
```

Example: what will be output of given code?



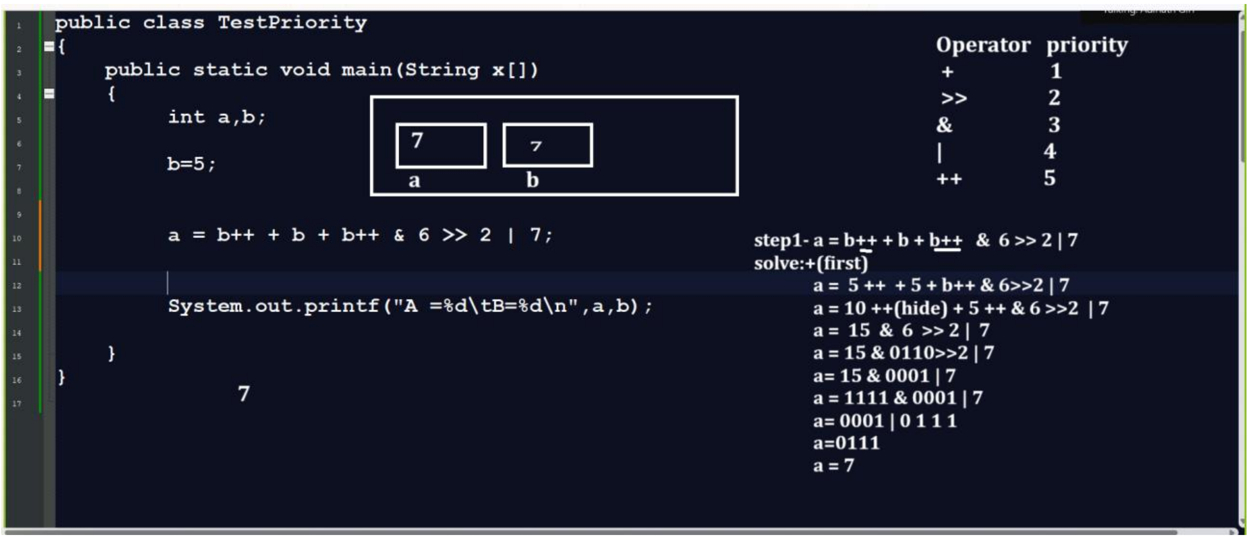
```
public class TestPriority
{
    public static void main(String x[])
    {
        int a,b;

        b=5;

        a = b++ + ++b;

        System.out.printf("A =%d\tB=%d\n",a,b);

    }
}
```

Output
A=12 B=7

6
a= b++ + ~~+~~+b;
step1: a= b++ + b;
12 a= 6 ++ +6   6+6 = 12
a= b++ 7

Note: if we think about above code a=b++ + ++b; here we have three operators in equation ++(pre) , +(addition) and ++(post) here ++ pre has higher priority so first ++b get executed shown in step1 and after that + has second priority as per our example so b++ +b here b + b execute first shown in second step2 and last ++(post) get executed means b++ executed so the last value of b was 6 then 6++ i.e 6+1 = 7 and this value get updated in variable b shown in step3

Q. What will be output of given code?



```
public class TestPriority
{
    public static void main(String x[])
    {
        int a,b;

        b=5;


        a = b++ + b + b++ & 6 >> 2 | 7;


        System.out.printf("A =%d\tB=%d\n",a,b);

    }
}
```

Operator	priority
+	1
>>	2
&	3
|	4
++	5

step1- a = b++ + b + b++ & 6 >> 2 | 7
solve:+(first)
a = 5 ++ + 5 + b++ & 6>>2 | 7
a = 10 ++(hide) + 5 ++ & 6 >>2 | 7
a = 15 & 6 >> 2 | 7
a = 15 & 0110>>2 | 7
a= 15 & 0001 | 7
a = 1111 & 0001 | 7
a= 0001 | 0 1 1 1
a=0111
a = 7