

How to Create Application using JAVA

If we want to create application using Java we have some important steps.

1. Installed the JDK

Q. What is JDK?

JDK stands for Java Development kit.

Def: JDK is a software cluster or it is group of software which provide environment to us to create and execute java application

it contain like as 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 source code?

Code written by programmer called as source code.

Q. What is byte code?

Byte code is intermediate format which easily convert in machine code with the help of JVM.
or byte code is machine understandable code.

Q. What is JVM?

JVM stands for Java Virtual machine and it is application present in JDK or in JRE which help us to Read byte code generated by compiler and put in ram and process on byte code like as verification of byte code, loading of byte code, allocation memory for object at run time, perform garbage collection etc

Q. Why Java Design byte code or what is benefit of byte code?

Byte is platform independent code

Q. What is platform?

Here platform indicate operating system means byte code can execute on any operation system without taking support any third party tool so java is platform independent language.

Example: if execute java code on windows then we can execute it on linux or any other Os it is platform independent code.

How to install the JDK

<https://www.oracle.com/in/java/technologies/downloads/#jdk22-windows>

download the .exe file and double click on it and install like as normal software

Note: when we installed your JDK then automatically we have folder on c drive program files

java\jdk-version

2. Create Sample Application

Note: If we want to create application or write code in java we have number of tools and editors

Editors: notepad, notepad++, WordPad, sublime text etc

Tools: eclipse, spring tool suite, intelliJ IDE etc

If we want to write code in java we required to write minimum following code or we have some generalize format for writing java code.

Generalize format of java code.

access specifier class classname

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

Example

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

Code Descriptions

public class AugFirst

public: public is access specifier

Q. What is access specifier?

Access specifier is the some keywords which are used for apply restrictions on class and its member.

There are four types of access specifier in java

public : public access specifier means we can access member within class or outside of class within same package as well as outside of package.

private: private means member cannot access outside of class means can access within class only.

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

default: default means member can access outside of class but within same package.

class: class is keyword which is used for class declaration purpose

AugFirst: it is class name and user can give any name to his class.

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

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

Meaning

System is class and out is static reference of PrintStream class. System and PrintStream maintain HAS-A relationship between and println() is overloaded method for display output on output screen.

3. Save Application:

If we want to save java program then save in bin folder where JDK install and give class name and filename same with .java extension.

4. Compile Application

Q. What is compilation?

Compilation is process which is used for convert your source code in to byte code.

How to compile java program

If we want to compile java program then we have following steps.

a. Open command prompt

start menu --- search --- command prompt

b. go where java file save

```
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

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

Note: if we think about above screen shot we can say our program get compile successfully.

if your program get compile successfully then java compiler create new file automatically with extension of .class and in this file contain your byte code means after compilation of code we have two files

AugFirst.java and AugFirst.class

AugFirst.java – contain source code

AugFirst.class- contain byte code.

5. Run the application: if we want to run java application then we have following command

java filename

Example: java AugFirst

Data type in JAVA

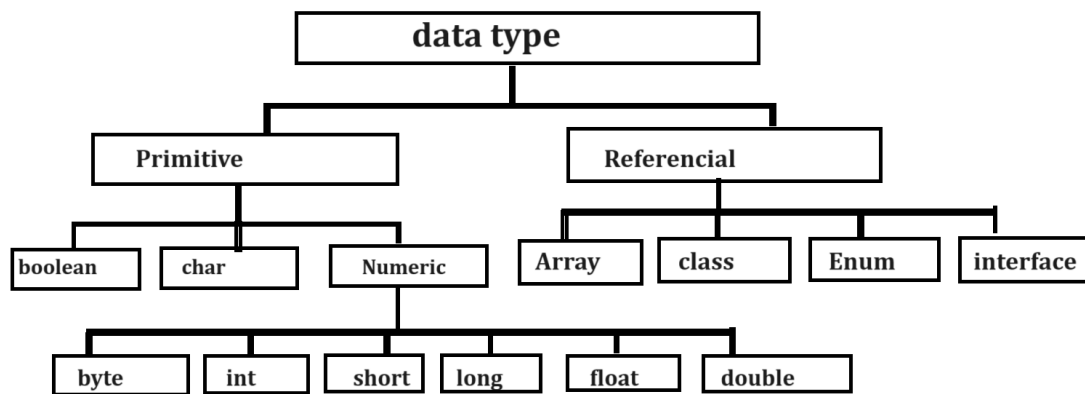
Q. What is Data type?

Data type means to decide what kind value or information we want to use in code called as data type.

There are two types of data type in JAVA

1. Primitive data type: primitive data type means those data type provided by java to us or inbuilt data type called as primitive data type and those data type variable unable to store address of memory and able to store only values called as primitive type of data type.

2. Non primitive or referential data type: those data type is able to store the address of memory and may be define user called as referential data type.



Now we want to discuss about the primitive data type.

Integer data type

If we want to work with any data type we should have to know four important points.

How to use in code: int

How much memory required: 4 byte

Q. why 4 byte?

This is not fix this dependent on compiler here java language use the JIT compiler and JIT required 4 byte memory for integer so it required 4 byte in future if Java language change compiler then There is possibility integer range may be vary.

Can we see the memory size of data type in JAVA?

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

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

Range or Calculation Capacity

-2147483648 to 2147483647

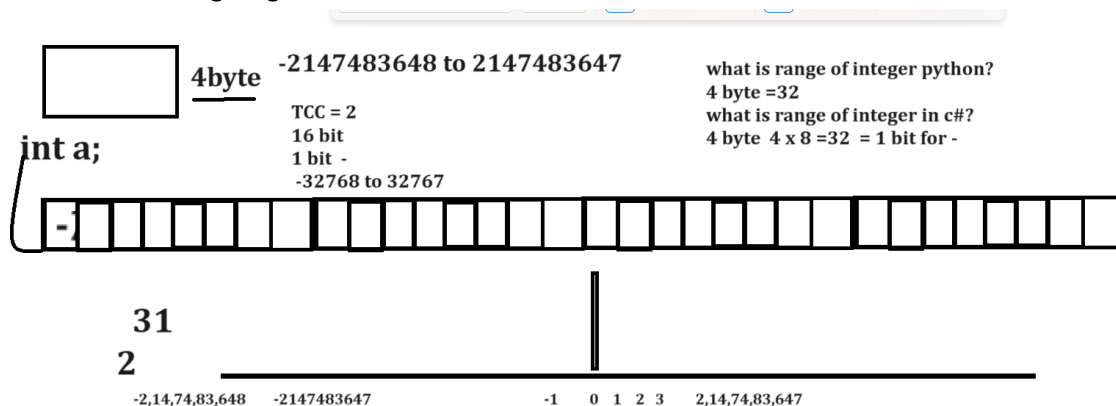
How we can calculate range?

If we think about above range int required 4 byte of memory means 32 bit and 1 bit required for 1 minus sign means we 31 bit for value so we have formula

$nbit-1$

2

shown in following diagram



Format specifier: format specifier indicate the type of data type in printf() statement

%d - %d represent integer in System.out.printf()

Example with source code

```
public class MemorySizeApp
{
    public static void main(String x[])
    {
        int a=100;
        System.out.printf("A = %d\n",a);
    }
}
```

long:

How to use in code: long

Memory required: 8 byte

Range:

63

2

Format specifier: %d

byte data type

How to use code: byte

Memory required: 1 byte

Range: -128 to 127

Format specifier: %d

Short integer data type

How to use in code: short

Memory required: 2 byte

Range: -32768 to 32767

15

2

Format specifier: %d

float data type

float data type is used for store the value with decimal point.

There are two types of float data type

float

How to use in code: float

Memory required: 4 byte

Range: 38 38

-3.14 x 10 to 3.14 x 10

Format specifier: %f

double:

How to use in code: double

Memory required: 8 byte

Range:

308 308

-1.7 x 10 to 1.7 x 10

Format specifier: %lf

Character Data type

We can use character data type for store alpha numeric values.

How to use in code: char

Memory required: 2 byte

Example with source code

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

Range: -32768 to 32767

Format specifier: %c

boolean: boolean data type is used for store true or false values.

How to use in code: boolean

Memory required : 1 byte

Format specifier: %b

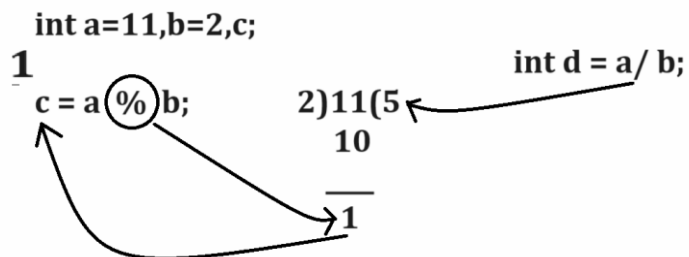
Operators in JAVA

Operators are the some symbols to perform some operations

Types of Operator in JAVA

1. Arithmetic Operator: Arithmetic operator is used for perform some arithmetic operations.

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



Example with source code

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

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

Operator	Meaning
=	assignment

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

Operator	Meaning
<	Less than
>	Greater Than
==	Equal to
!=	Not Equal
<=	Less than equal
>=	Greater than equal.

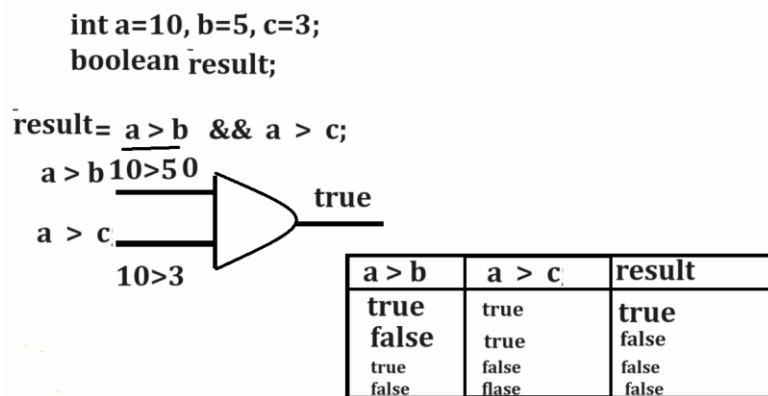
Note: Relational operator always return boolean value or return type of relational operator is boolean

```
public class TestRelationalApp
{
    public static void main(String x[])
    {
        int a=11,b=2;
        boolean result;
        true result = a > b;
        System.out.println("Result is "+result);
    }
}
```


4. Logical Operator: Logical operator is used for combine more than one conditions and marks it as single condition and generate result.

Operator	Meaning
&&	Logical AND If all conditions true then condition is true otherwise condition is false
	Logical OR If any one 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.



Example with source code

```
public class TestLogicalApp
{ public static void main(String x[])
{   int a=10,b=5,c=3;
    boolean result;
    result = a > b && a>c;
    System.out.println("Result is "+result);

}
}
```

Output

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

C:\Program Files\Java\jdk-22\bin>java TestLogicalApp
Result is true
```

5. Increment and Decrement operator

Increment and decrement operator is used for increase the value by 1 and decrease the value by 1 and it is short cut +1

There are two types of operator in Increment or Decrement

Operator	Meaning
++	Increment By 1
--	Decrement By 1

There are two types of increment and decrement

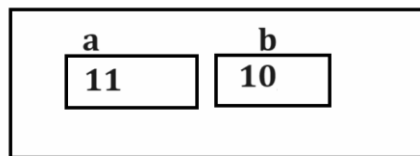
a. pre increment and pre decrement: pre increment means first we perform operation and shift value after operation.

Syntax: op variablename;

b. post increment and post decrement: post increment means first we initialize value at left hand side and after that perform operation.

Syntax: variablename op;

```
int a=10,b;  
b = a++;
```



```
System.out.printf("A=%d\tB=%d\n",a,b);  
A=11 B=10
```

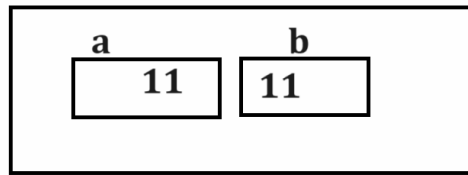
Example with source code

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

Example of pre increment

```
int a=10,b;
```

```
b = ++a;  
11
```



```
System.out.printf("A=%d\tB=%d\n",a,b);  
A=11 B=11
```

Example with source code

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

6. Conditional Operator

Conditional operator is used for check the condition

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

Exp1 is always condition and exp2 and exp3 are the options if exp1 is true then execute exp2 and if exp1 is false then execute exp3

```
public class TestConditionalApp  
{  
    public static void main(String x[])  
    {  
        int a=11,b=2;  
        A is Greater //exp1          exp2          exp3  
        String result = a > b ? " A is Greater " : " B is Greater ";  
        System.out.println(result);  
    }  
}
```

Example:

```
public class CheckConditionApp  
{  
    public static void main(String x[])  
    {  
        int a=11,b=2,c;  
        String result= a>b ? "A is Greater" : "B is Greater";  
    }  
}
```

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

Example for check number is even or odd

```
public class CheckConditionApp
{
    public static void main(String x[])
    {
        int a=11,c;
        String result= a%2==0 ? "Number is Even" : "Number is Odd";
        System.out.println(result);
    }
}
```

Output

```
C:\Program Files\Java\jdk-22\bin>javac CheckConditionApp.java
C:\Program Files\Java\jdk-22\bin>java CheckConditionApp
Number is Odd
```

Note: if we think about above code we have fix value a=11 for check number status even or odd but we want to accept the input from keyboard and after that decide number is even or odd.

How to accept the input from keyboard using JAVA

If we want to accept input from keyboard using JAVA we have two ways.

1. Using Command Line Argument
2. Using Scanner class

Q. What is command line argument?

Command line argument is parameter present in main of string array and it is used for accept input from keyboard in the form of string and it is infinite string array means we can accept n number of input value using command line argument but the first input of command line argument is at position of 0.

```
public class TestConditionalApp
{
    public static void main(String x[])
    {
        // command line arguments.
    }
}
```

Example using command line argument

We want to accept the input of two values from keyboard at program run time and calculate its addition.

```
public class AddApp
{
    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\jdk-22\bin>javac AddApp.java
AddApp.java:6: error: incompatible types: String cannot be converted to int
    a=x[0]; //first input
    ^
AddApp.java:7: error: incompatible types: String cannot be converted to int
    b=x[1]; //second input
    ^
2 errors
```

Note: if we think about above code we get two errors incompatible types because we have string for input and we want to accept the input of type integer and string cannot store directly in integer so compiler will generate compile time error to us.

```
public class AddApp
{
    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);
    }
}
```

variable a and b is type of integer.

Here x[0] and x[1] is type of string.

x[0] is string cannot store in int a and x[1] is string cannot store in int b 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 to 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 string to integer

How to convert String value to integer value in JAVA

If we want to convert String value to integer value in java we have to use following syntax

Syntax: `int variable = Integer.parseInt(String)`: Here Integer is class and parseInt() is function which is used for convert your string to integer in JAVA.

Note: if we want to identify class in Java then every class first letter is capital as per the standard of Java and if your class form by using more than one words then every word first letter is capital.

Example: Scanner, DataInputStream and if we want to identify the function in java then every function first letter is small alphabet and if function form by using single word then all letter of function must be small and if function form by more than one word then first word first letter is small and after that every word first letter is capital.

Example: `int read (), String readLine()`

Example: `int a =Integer.parseInt(x[0]);`

Example for convert String to float

`float variable=Float.parseFloat(String)`: this function is used for convert string value to float value.

Example for convert string to double

`double variable=Double.parseDouble(String)`: this function is used for convert string value to double value.

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

C:\Program Files\Java\jdk-22\bin>java AddApp
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 0 out of bounds for length 0
at AddApp.main(AddApp.java:6)

x[0]	x[1]
100	200

C:\Program Files\Java\jdk-22\bin>java AddApp
Addition is 300

x[0]	x[1]
100	

C:\Program Files\Java\jdk-22\bin>java AddApp
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 1 out of bounds for length 1
at AddApp.main(AddApp.java:8)

Example: WAP to input the quantity and rate of product and calculate its total bill

```
public class BillingApp
{
    public static void main(String x[])
    {
        int qty,rate,total;
        qty=Integer.parseInt(x[0]);
        rate=Integer.parseInt(x[1]);
        total=qty*rate;
        System.out.printf("Total is %d\n",total);
    }
}
```

```
}  
}
```

Example: WAP to input the radius from circle and calculate its area.

```
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);  
  }  
}
```

Scanner class

Scanner class is used for accept the input from keyboard at program run time on new line.

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 and it is like as header file in c language and if we want to add package in code we have to use import keyword.

Syntax: `import package.*;` this statement indicate we can import all member from package in application.

or

import package.classname: this statement indicate 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 its method to work with Scanner

If we want to work with Scanner we have some inbuilt method provided by Scanner class to us.

int nextInt(): this method is used for accept the input of type integer

float nextFloat(): this method is used for accept the input of type float

double nextDouble(): this method/function is used for accept input of type double

long nextLong(): this method is used for accept the input type of long

short nextShort(): this method is used for accept the input of type short

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

etc

Note: above methods are the members of Scanner class so if we want to use any method in code we have to use following syntax

Syntax: **datatype variablename**= scannerobj.methodname();

Example: we want to accept two values as input and calculate its addition.

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

Example: WAP to input name,id and salary of employee and display it

```
import java.util.*; //step1- import package
public class EmployeeApp
{
    public static void main(String x[])
    {
        Scanner xyz = new Scanner(System.in); //step2- create object of scanner
        String name;
        int id;
        long sal;
        System.out.println("Enter name id and salary of employee");
        name=xyz.nextLine(); //accept input from keyboard.
        id=xyz.nextInt();
        sal=xyz.nextLong();
        System.out.printf("Name is %s\n",name);
        System.out.printf("Id is %d\n",id);
        System.out.printf("Salary is %d\n",sal);
    }
}
```



```

}
}

```

Example: WAP to input basic salary of employee and calculate its total salary using a following terms
da=30% and hra=30%

bs=1000

da=300

hra=300

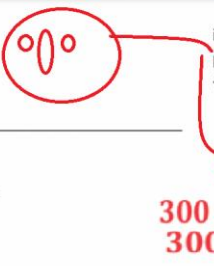
total= bs + da + hra;

Example: WAP to input basic salary of employee and calculate its total salary using a following terms
da=30% and hra=30%

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

Steps

1. input basic salary
2. calculate its da and hra
Note: need to implement percentage logic
3. Calculate total salary
4. show the total salary.



```

import java.util.*; //step1- package import
public class GrossSalApp
{
    public static void main(String x[])
    {
        Scanner xyz = new Scanner(System.in); //step2- object creation
        int bs, da, hra, total;
        System.out.println("Enter basic salary of employee");
        bs = xyz.nextInt(); //step3- accept input
        da = bs * 30 / 100; // 1000*30/100=300
        hra = bs * 30 / 100; // 1000*30/100=300
        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

bs	da	hra	total
1000	300	300	1600

Example:

WAP to input two values from keyboard and perform swapping on it without using third variable

```
import java.util.*; //step1;
```

```
public class SwapApp
```

```
{
```

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

```
    { Scanner xyz = new Scanner(System.in); //step2
```

```
        int a, b;
```

```
        System.out.println("Enter two values");
```

```
        a = xyz.nextInt(); //step3- accept input
```

```
        b = xyz.nextInt();
```

```
        System.out.printf("Before Swapping\n");
```

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

```
        /*
```

```
        Logic1:
```

```
        a = a * b;
```

```
        b = a / b;
```

```
        a = a / b;
```

```
        Logic2:
```

```
        a = a + b;
```

```
        b = a - b;
```

```
        a = a - b;
```

```

*/
a=a^b;
b=a^b;
a=a^b;
System.out.printf("After swapping\n");
System.out.printf("A=%d\tB=%d\n",a,b);

}
}

```

10)123(12
120
—
3

10)12(1
10
—
2

10)1(0
—
1

no=123

10)123(12
120
—
3

10)12(1
10
—
2

10)1(0
—
1

3*100+2*10+1*1= 321

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

no	rem	rev
12	1	321

or

10)123(12
120
—
3

10)12(1
10
—
2

10)1(0
—
1

no=123

10)123(12
120
—
3

10)12(1
10
—
2

10)1(0
—
1

3*100+2*10+1*1= 321

no=123;

321

3*100 + ~~120~~10 2*10 + 1

no = (no%10)*100 + ((no/10)%10)*10 + (no/100);

10)123(12
120
—
3

10)123(12
120
—
2

100)123(1

321

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

no=1234

no 4231

last 4000

first 1

4000
+23*10

4 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

10)1234(123
1230
—
4

100)123(1
100
—
23

1234=one thousand two hundred thirty four

4231 = four thousand

Example with source code

```
import java.util.*;
public class FirstLastSwapApp
{
    public static void main(String x[])
    { Scanner xyz = new Scanner(System.in);
        int no,last,first;
        System.out.println("Enter number");
        no=xyz.nextInt();
        System.out.printf("Before Swap %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 swap %d\n",no);

    }
}
```

Chef has to travel to another place. For this, he can avail any one of two cab services.

- The first cab service charges XX rupees.
- The second cab service charges YY rupees.

Chef wants to spend the **minimum** amount of money. Which cab service should Chef take?

Input Format

- The first and only line of each test case contains two integers XX and YY - the prices of first and second cab services respectively.

Output Format

For each test case, output FIRST if the first cab service is cheaper, output SECOND if the second cab service is cheaper, output ANY if both cab services have the same price.

You may print each character of FIRST, SECOND and ANY in uppercase or lowercase (for example, any, aNy, Any will be considered identical).

Constraints

$1 \leq T \leq 100$

$1 < X < Y \leq 100$

Sample 1:

Input Output:

```
3
30 65    First
42 42    Any
```

Output

```
import java.util.*;
public class CabServiceApp
{ public static void main(String x[])
  { Scanner xyz = new Scanner(System.in);
    int xx,yy;
    System.out.println("Enter the first and second cab service charges");
    xx=xyz.nextInt();
    yy=xyz.nextInt();

    String result = xx==yy ? "Any": (xx>yy?"second ":"first");

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

7. Bitwise Operator

Bitwise operator is used for perform operation by using bits

Note: if we want to work with bitwise operator we should have to know the some important points.

1. Number System**2. Conversion of decimal number to binary or binary number to decimal or any others.****3. Binary AND, OR, XOR operation****4. One's Complement and 2's complement****Q. What is Number System?**

Number system is technique which is used for represent how number should store in computer memory called as Number System.

If we want to represent number in computer we have four ways.

1. Decimal Number: decimal number means a number contain 0 to 9 values called as decimal and the base of decimal is 10.

Q. what is base?

Base means number of digit present in number system and it is used for identify number.

Example: (123456)

10 – base

Example: (1010101)

10

2. Binary Number: Binary number system means it contain only two digits i.e 0 & 1 so the base of binary number system is 2.

Example: (010101)

2 – binary number.

3. Octal Number: Octal number system means it contain 0 to 7 digits so the base of octal is 8.

Example:

(1234)

8 - base

4. Hexa decimal: Hexa decimal number system means it contain value between 0 to 15 and from 10 it contain 10=A,11=B,12=C,13=D,14=E and 15=F and the base of hexa decimal is 16

Example: (1234AF)

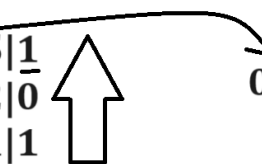
16

How to convert decimal number in to binary

If we want to perform conversion decimal to binary we required to divide your decimal number by 2 and calculate its reminder and put all reminders at right hand side and pick all reminders from bottom to top

we want to convert 5 decimal value in to binary.


2		5		1
2		2		0
2		1		1



0101: it is binary of decimal 5.

Example: we want to convert 12 decimal values in to binary

2		12		0
2		6		0
2		3		1
2		1		1



1100 - it is binary of decimal 12

How to convert binary to decimal value

If we want to convert binary to decimal values we have to multiply every binary digit by 2 and arrange its index from right hand side and start first index from 0 to n-1

Example: Convert 1100 binary value in to decimal value.

$$\begin{array}{cccc}
 3 & 2 & 1 & 0 \\
 1 \times 2 + 1 \times 2 + 0 \times 2 + 0 \times 2 \\
 1 \times 8 + 1 \times 4 + 0 \times 2 + 0 \times 1 \\
 8 + 4 + 0 + 0 = 12
 \end{array}$$

Binary AND Operation

Binary AND Operation indicate if first input is 1 and second is 1 then output is 1 otherwise 0.

Example:

first input: 1 0 1 0
second input: 0 1 0 1

true table of binary AND

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

0 0 0 0 - it is result of binary AND operation

Binary OR Operations

Binary OR Operation means if first input is 1 and second is 0 or first is 1 and second is 1 or first 0 and second is 1 then 1 otherwise 0 means if first is 0 and second is 0 then result is 0.

truth table of binary OR Operation

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

first input: 0 1 0 1
second input: 1 0 1 0

1 1 1 1

Result of your binary OR Operation

Binary XOR Operation

If we think about XOR operation on two different bit and if we have same first input and same second input then result is 0 and if we have different inputs then 1

Truth table of XOR operator.

First input	second input	result
0	0	0
0	1	1
1	0	1
1	1	0

First input: 0 1 0 1
Second input 1 1 0 1

1 0 0 0

Result of XOR Operations.

One's complement is negation operation on binary digit means in the case of one's complement we convert 1 to 0 and 0 to 1.

Original binary :	0	1	0	1
One's complement value :	1	0	1	0

Two's complement

if we want to perform two's complement on any binary digit then first we need to convert binary value in to one's complement and after that we required to add 1 at last in one's complement value.

Original binary :	0	1	0	1
One's complement value :	1	0	1	0

1 0 1 1 ⁺¹ 2's complement

Now we want to work on bitwise operator

Operator	Meaning
	Bitwise OR: this operator convert decimal values in binary digit and perform bitwise OR operation on it and generate result in binary and again convert in decimal and store in left hand side variables.
&	Bitwise AND
<<	Bitwise Left shift operator
>>	Bitwise right shift operator
>>>	Bitwise right shift with unsign value
^	XOR Operator
~	One's complement operator

```
int a=5,b=6,c;
```

```
c = a | b;
```

$c = \bar{5} \mid 6$; the meaning of this statement is internally

$$c = \begin{array}{r} 0101 \\ | 0110 \\ \hline 0111 \end{array}$$
$$\begin{array}{rcccc} & 3 & 2 & 1 & 0 \\ 0 \times 2^3 & + 1 \times 2^2 & + 1 \times 2^1 & + 1 \times 2^0 & \\ 0 \times 8 & + 1 \times 4 & + 1 \times 2 & + 1 \times 1 & \\ 0 & + 4 & + 2 & + 1 & = 7 \end{array}$$

Example with source code

```
public class BitWiseORApp
{
    public static void main(String x[])
    {
        int a=5,b=6,c;
        c=a | b;
        System.out.printf("C is %d\n",c);
    }
}
```

Example with source code

int a=13,b=14,c;

c = a | b;

c = 13 | 14;

c= 1 1 0 1
| 1 1 1 0

1 1 1 1

 3 2 1 0
1 x 2 + 1 x 2 + 1 x 2 + 1 x 2
1 x 8 + 1 x 4 + 1 x 2 + 1 x 1 = 15

Example with source code

```
public class BitWiseORApp
{ public static void main(String x[])
{   int a=13,b=14,c;
    c=a | b;
    System.out.printf("C is %d\n",c);
}
}
```

Bitwise & operator: Bitwise & operator can convert decimal values in to binary and perform bitwise AND operation on it and again convert result in decimal format.

int a=5,b=4,c;

4
c = a & b;

c = 5 & 4
c = 0 1 0 1
& 0 1 0 0

 3 2 1 0
0 x 2 + 1 x 2 + 0 x 2 + 0 x 2
0 x 8 + 1 x 4 + 0 x 2 + 0 x 1 = 4

Example with source code

```
public class BitwiseAndApp
```

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

Example

```
int a=14 b=13;
int c = a & b;
```

12

c = 1 1 1 0
 & 1 1 0 1

 1 1 0 0

3 2 1 0
 $1 \times 2 + 1 \times 2 + 0 \times 2 + 0 \times 2$
 $1 \times 8 + 1 \times 4 + 0 + 0 = 12$

Example with source code

```
public class BitwiseAndApp
```

```
{
    public static void main(String x[])
    {
        int a=14,b=13,c;
        c=a&b;
        System.out.printf("C is %d\n",c);
    }
}
```

(<<) Left shift operator:

This operator can shift the bit from right hand side to left hand side and 0's at right hand side.

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

20

5

4byte

0 1 0 1

0 1 0 1 0 0

4 3 2 1 0
 $1 \times 2 + 0 \times 2 + 1 \times 2 + 0 \times 2 + 0 \times 2$
 $1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 0 \times 1$
 $16 + 0 + 4 + 0 + 0 = 20$

```
public class BitwiseAndApp
```

Example:

Example with source code

>>(Right shift operator): this operator can shift bit at right hand side and 0's at left hand side.

[illegible]

Example with source code

```
public class BitwiseAndApp
{
    public static void main(String x[])
    {
        int a=5,b;
        b=a>>2;
        System.out.printf("B is %d\n",b);
    }
}
```

Output

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac BitwiseAndApp.java
C:\Program Files\Java\jdk1.8.0_291\bin>java BitwiseAndApp
B is 1
C:\Program Files\Java\jdk1.8.0_291\bin>
```

Bitwise XOR (^) operator

int a=5,b=6,c;

c = a ^ b;

0	1	0	1
0	1	1	0
<hr/>			
0	0	1	1

3 2 1 0

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

$0 + 0 + 2 + 1 = 3$

Example with source code

```
public class BitwiseAndApp
{
    public static void main(String x[])
    {
        int a=5,b=6,c;
        c=a^b;
        System.out.printf("C is %d\n",c);
    }
}
```

Swapping example using XOR operator

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

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

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

    a=a^b;

    b=a^b;

    a=a^b;

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

}
```

~ (One's complement): this is the negation operation on binary digit means if your binary digit is 0 then convert in 1 and if binary digit is 1 then converts in 0.

[illegible]

Example with source code

```
public class BitwiseAndApp
```

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

Output

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac BitwiseAndApp.java
C:\Program Files\Java\jdk1.8.0_291\bin>java BitwiseAndApp
B is -6
C:\Program Files\Java\jdk1.8.0_291\bin>
```

Example :

```

public class BitwiseAndApp
{
    public static void main(String x[])
    {
        int a=1000,b;
        b=~a;
        System.out.printf ("B is %d\n",b);
    }
}

```

Operator priority or Operator precedence

Precedence	Operator	Description	Association
1	(), [], . (dot)	Parenthesis , square bracket and dot operator	Left to right
2	++,--(pre)	Increment and decrement (pre)	Right to left
3	- (unary) ! ~ type	unary minus unary logical negation one's complement type	Right to left
4	* / %	Multiplication Division Reminder or mod	Left to right
5	+ -	Addition Subtraction	Left to right
6.	<< >> >>>	Bitwise left shift Bitwise right shift with sign Bitwise right shift with zero extension	Left to right
7	< <= > >= Instanceof	Less than Less than equal Greater than Greater than equal Instanceof operator	Left to right
8	== !=	Relational Equal operator Relational Not equal operator	Left to right
9	&	Bitwise AND	Left to right
10	^	Bitwise XOR	Left to right
11		Bitwise OR	Left to right
12	&&	Logical AND	Left to right
13		Logical OR	Left to right

14	? :	Ternary operator and colon	Right to left
15.	++ ,--(post)	Post increment and decrement	Left to right
16	=, +=,-=,*=,/= etc	Short hand operator	Right to left

Example

```
public class PAPP
```

```
{
```

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

```
    {
```

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

```
        a= b+5*3/4*5+(9-3)*6/2+7-8*3/2+2;
```

```
        //step1= a=5+5*3/4*5+6*6/2+7-8*3/2+2
```

```
        //step2= a=5+15/4*5+6*6/2+7-8*3/2+2
```

```
        //step3= a=5+3*5+6*6/2+7-8*3/2+2
```

```
        //step4= a=5+15+36/2+7-8*3/2+2
```

```
        //step5= a=5+15+18+7-8*3/2+2
```

```
        //step6 a=5+15+18+7-24/2+2
```

```
        //step7 a=5+15+18+7-12+2
```

```
        //step8 a=20+18+7-12+2
```

```
        //step9 a=38+7-12+2
```

```
        //step10 a=45-12+2
```

```
        //step11 a=45-12+2
```

```
        //step12 a=33+2
```

```
        //step13 a=35
```

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

```
    }
```

```
}
```

Example with source code

```
public class PAPP
```

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

```
    {        int a=5,b;
```

```
        b= a + 5 & 6 >> 2 + 3 ^ 7;
```

```
        // a=10 & 6 >> 2 + 3 ^ 7
```

```
        //a= 10 & 6 >> 5 ^ 7
```

```
        //a=10 & 0 ^ 7
```

```
        // a= 0 ^ 7
```

```
// a= 0 0 0 0
//  0 1 1 1
//  0 1 1 1
// b=7

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

Example