**Java program to swap two numbers**

This java program swaps two numbers using a temporary variable. To swap numbers without using extra variable see another code below.

**Swapping using temporary or third variable**

**import** java.util.Scanner;

**class** SwapNumbers

{

**public** **static** **void** main(String args[])

{

**int** x, y, temp;

System.out.println("Enter x and y");

Scanner in = **new** Scanner(System.in);

x = in.nextInt();

y = in.nextInt();

System.out.println("Before Swapping**\n**x = "+x+"**\n**y = "+y);

temp = x;

x = y;

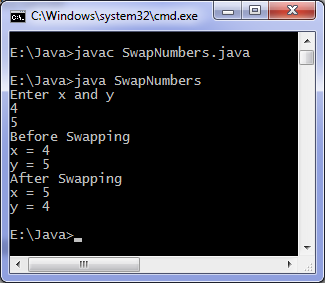
y = temp;

System.out.println("After Swapping**\n**x = "+x+"**\n**y = "+y);

}

}

[Swap numbers](http://www.programmingsimplified.com/executable/java/SwapNumbers.class) program class file.

Output of program:  


**Swapping without temporary variable**

**import** java.util.Scanner;

**class** SwapNumbers

{

**public** **static** **void** main(String args[])

{

**int** x, y;

System.out.println("Enter x and y");

Scanner in = **new** Scanner(System.in);

x = in.nextInt();

y = in.nextInt();

System.out.println("Before Swapping**\n**x = "+x+"**\n**y = "+y);

x = x + y;

y = x - y;

x = x - y;

System.out.println("After Swapping**\n**x = "+x+"**\n**y = "+y);

}

}

# Java program to find largest of three numbers

This java program finds largest of three numbers and then prints it. If the entered numbers are unequal then "numbers are not distinct" is printed.

## [Java programming](http://www.cgstatic.info/code/r.php?r=yahoo%7CJava%2520programming&t=39&did=39&uid=0&type=bl&subid=raftxXYZcay%24&rkw=Java+programming&rurl=http%3A%2F%2Fwww.programmingsimplified.com%2Fjava%2Fsource-code%2Fjava-program-largest-of-three-numbers&domain=programmingsimplified.com&lnktype=10&v=0.126&browser=Firefox_44&country=IN&_=1458448407856) source code

**import** java.util.Scanner;

**class** LargestOfThreeNumbers

{

**public** **static** **void** main(String args[])

{

**int** x, y, z;

System.out.println("Enter three integers ");

Scanner in = **new** Scanner(System.in);

x = in.nextInt();

y = in.nextInt();

z = in.nextInt();

**if** ( x > y && x > z )

System.out.println("First number is largest.");

**else** **if** ( y > x && y > z )

System.out.println("Second number is largest.");

**else** **if** ( z > x && z > y )

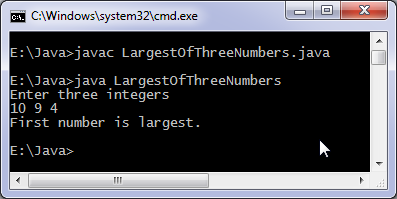
System.out.println("Third number is largest.");

**else**

System.out.println("Entered numbers are not distinct.");

}

}

Output of program:  


# Enhanced for loop java

Enhanced for loop java: Enhanced for loop is useful when scanning the array instead of using for loop. Syntax of enhanced for loop is:  
**for** (data\_type variable: array\_name)  
Here array\_name is the name of array.

## Java enhanced for loop integer array

**class** EnhancedForLoop {

**public** **static** **void** main(String[] args) {

**int** primes[] = { 2, 3, 5, 7, 11, 13, 17, 19, 23, 29};

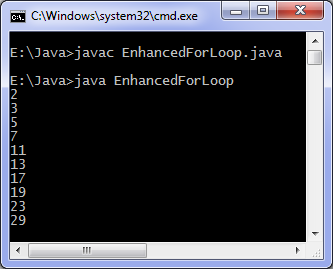
**for** (**int** t: primes) {

System.out.println(t);

}

}

}

Output of program:  


## Java enhanced for loop strings

**class** EnhancedForLoop {

**public** **static** **void** main(String[] args) {

String languages[] = { "C", "C++", "Java", "Python", "Ruby"};

**for** (String sample: languages) {

System.out.println(sample);

}

}

}

# Java program to find factorial

This java program finds factorial of a number. Entered number is checked first if its negative then an error message is printed.

## Java programming code

**import** java.util.Scanner;

**class** Factorial

{

**public** **static** **void** main(String args[])

{

**int** n, c, fact = 1;

System.out.println("Enter an integer to calculate it's factorial");

Scanner in = **new** Scanner(System.in);

n = in.nextInt();

**if** ( n < 0 )

System.out.println("Number should be non-negative.");

**else**

{

**for** ( c = 1 ; c <= n ; c++ )

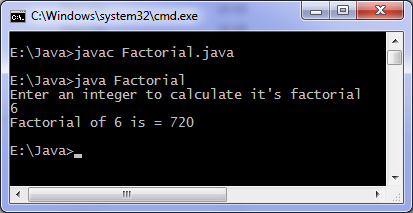
fact = fact\*c;

System.out.println("Factorial of "+n+" is = "+fact);

}

}

}

Output of program:  


You can also find factorial using recursion, in the code fact is an integer variable so only factorial of small numbers will be correctly displayed or which fits in 4 bytes. For large numbers you can use long data type.

## Java program for calculating factorial of large numbers

Above program does not give correct result for calculating factorial of say 20. Because 20! is a large number and cant be stored in integer data type which is of 4 bytes. To calculate factorial of say hundred we use BigInteger class of java.math package.

**import** java.util.Scanner;

**import** java.math.BigInteger;

**class** BigFactorial

{

**public** **static** **void** main(String args[])

{

**int** n, c;

BigInteger inc = **new** BigInteger("1");

BigInteger fact = **new** BigInteger("1");

Scanner input = **new** Scanner(System.in);

System.out.println("Input an integer");

n = input.nextInt();

**for** (c = 1; c <= n; c++) {

fact = fact.multiply(inc);

inc = inc.add(BigInteger.ONE);

}

System.out.println(n + "! = " + fact);

}

}

We run the above java program to calculate 100 factorial and following output is obtained.

Input an integer

100

100! = 93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916864000000000000000000000000

# Java program print prime numbers

This java program prints prime numbers, number of prime numbers required is asked from the user. Remember that smallest prime number is 2.

## Java programming code

**import** java.util.\*;

**class** PrimeNumbers

{

**public** **static** **void** main(String args[])

{

**int** n, status = 1, num = 3;

Scanner in = **new** Scanner(System.in);

System.out.println("Enter the number of prime numbers you want");

n = in.nextInt();

**if** (n >= 1)

{

System.out.println("First "+n+" prime numbers are :-");

System.out.println(2);

}

**for** ( **int** count = 2 ; count <=n ; )

{

**for** ( **int** j = 2 ; j <= Math.sqrt(num) ; j++ )

{

**if** ( num%j == 0 )

{

status = 0;

**break**;

}

}

**if** ( status != 0 )

{

System.out.println(num);

count++;

}

status = 1;

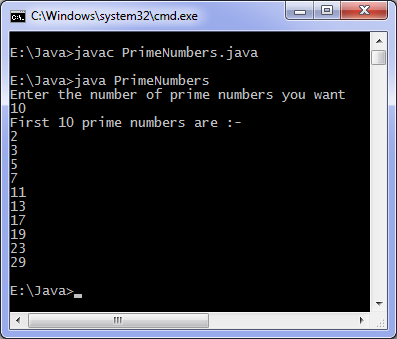
num++;

}

}

}

Download [Prime numbers](http://www.programmingsimplified.com/executable/java/PrimeNumbers.class) program class file.

Output of program:  


# Java program to [reverse a string](http://www.cgstatic.info/code/r.php?r=yahoo%7Creverse%2520a%2520string%2520%2520%2520&t=39&did=39&uid=0&type=bl&subid=raftxXYZcay%24&rkw=reverse+a+string&rurl=http%3A%2F%2Fwww.programmingsimplified.com%2Fjava%2Fsource-code%2Fjava-program-reverse-string&domain=programmingsimplified.com&lnktype=10&v=0.126&browser=Firefox_44&country=IN&_=1458448569012)

This java program [reverses a string](http://www.cgstatic.info/code/r.php?r=yahoo%7Creverses%2520a%2520string&t=39&did=39&uid=0&type=bl&subid=raftxXYZcay%24&rkw=reverses+a+string&rurl=http%3A%2F%2Fwww.programmingsimplified.com%2Fjava%2Fsource-code%2Fjava-program-reverse-string&domain=programmingsimplified.com&lnktype=10&v=0.126&browser=Firefox_44&country=IN&_=1458448569373) entered by the user. We use charAt method to extract characters from the string and append them in reverse order to reverse the entered string.

## [Java programming](http://www.cgstatic.info/code/r.php?r=yahoo%7CJava%2520programming&t=39&did=39&uid=0&type=bl&subid=raftxXYZcay%24&rkw=Java+programming&rurl=http%3A%2F%2Fwww.programmingsimplified.com%2Fjava%2Fsource-code%2Fjava-program-reverse-string&domain=programmingsimplified.com&lnktype=10&v=0.126&browser=Firefox_44&country=IN&_=1458448570379) code

**import** java.util.\*;

**class** ReverseString

{

**public** **static** **void** main(String args[])

{

String original, reverse = "";

Scanner in = **new** Scanner(System.in);

System.out.println("Enter a string to reverse");

original = in.nextLine();

**int** length = original.length();

**for** ( **int** i = length - 1 ; i >= 0 ; i-- )

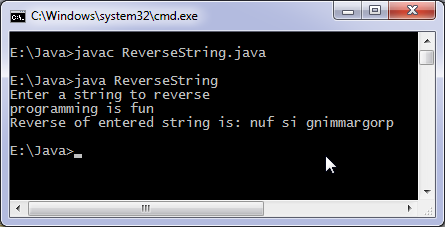
reverse = reverse + original.charAt(i);

System.out.println("Reverse of entered string is: "+reverse);

}

}

Download [Reverse string](http://www.programmingsimplified.com/executable/java/ReverseString.class) program class file.

Output of program:  


### Reverse string using StringBuffer class

**class** InvertString

{

**public** **static** **void** main(String args[])

{

StringBuffer a = **new** StringBuffer("[Java programming is](http://www.cgstatic.info/code/r.php?r=yahoo%7CJava%2520programming%2520is&t=39&did=39&uid=0&type=bl&subid=raftxXYZcay%24&rkw=Java+programming+is&rurl=http%3A%2F%2Fwww.programmingsimplified.com%2Fjava%2Fsource-code%2Fjava-program-reverse-string&domain=programmingsimplified.com&lnktype=10&v=0.126&browser=Firefox_44&country=IN&_=1458448571978) fun");

System.out.println(a.reverse());

}

}

StringBuffer class contains a method reverse which can be used to reverse or invert an object of this class.

# Java program to check palindrome

Java palindrome program: Java program to check if a string is a palindrome or not. Remember a string is a palindrome if it remains unchanged when reversed, for example "dad" is a palindrome as reverse of "dad" is "dad" whereas "program" is not a palindrome. Some other palindrome strings are "mom", "madam", "abcba".

## Java programming source code

**import** java.util.\*;

**class** Palindrome

{

**public** **static** **void** main(String args[])

{

String original, reverse = "";

Scanner in = **new** Scanner(System.in);

System.out.println("Enter a string to check if it is a palindrome");

original = in.nextLine();

**int** length = original.length();

**for** ( **int** i = length - 1; i >= 0; i-- )

reverse = reverse + original.charAt(i);

**if** (original.equals(reverse))

System.out.println("Entered string is a palindrome.");

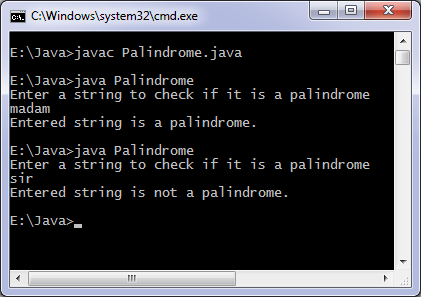
**else**

System.out.println("Entered string is not a palindrome.");

}

}

Download [Palindrome](http://www.programmingsimplified.com/executable/java/Palindrome.class) program class file.

Output of program:  


Another method to check palindrome:

**import** java.util.\*;

**class** Palindrome

{

**public** **static** **void** main(String args[])

{

String inputString;

Scanner in = **new** Scanner(System.in);

System.out.println("Input a string");

inputString = in.nextLine();

**int** length = inputString.length();

**int** i, begin, end, middle;

begin = 0;

end = length - 1;

middle = (begin + end)/2;

**for** (i = begin; i <= middle; i++) {

**if** (inputString.charAt(begin) == inputString.charAt(end)) {

begin++;

end--;

}

**else** {

**break**;

}

}

**if** (i == middle + 1) {

System.out.println("Palindrome");

}

**else** {

System.out.println("Not a palindrome");

}

}

}