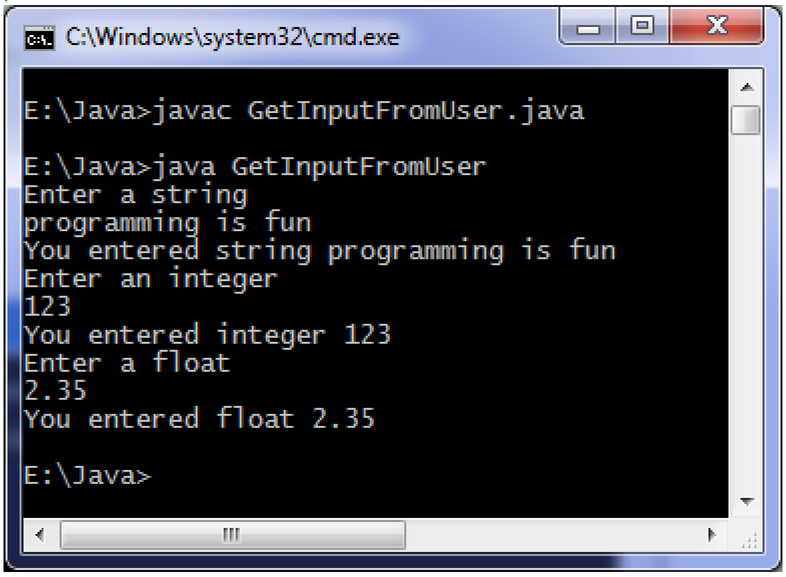
How to get input from user

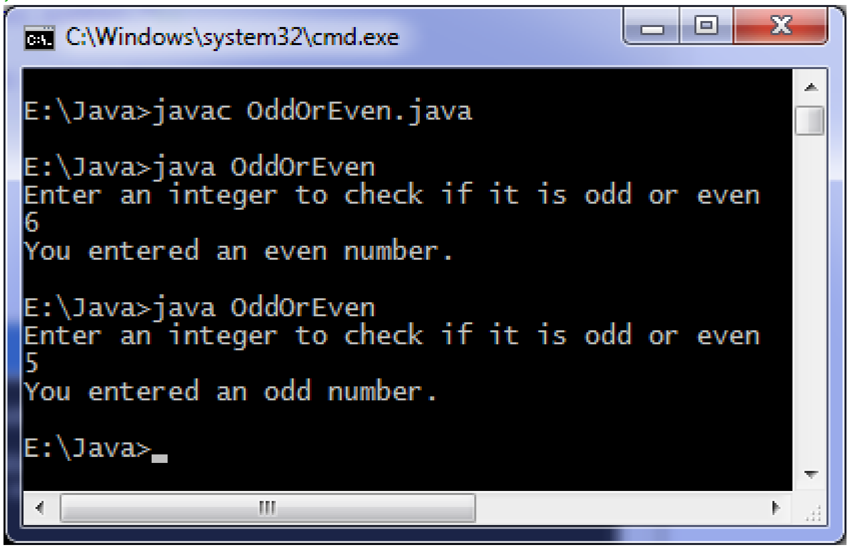
```java
import java.util.Scanner;

class GetInputFromUser
{
   public static void main(String args[])
   {
      int a;
      float b;
      String s;

      Scanner in = new Scanner(System.in);

      System.out.println("Enter a string");
      s = in.nextLine();
      System.out.println("You entered string "+s);

      System.out.println("Enter an integer");
      a = in.nextInt();
      System.out.println("You entered integer "+a);

      System.out.println("Enter a float");
      b = in.nextFloat();
      System.out.println("You entered float "+b);
   }
}
```



```
E:\Java>javac GetInputFromUser.java

E:\Java>java GetInputFromUser
Enter a string
programming is fun
You entered string programming is fun
Enter an integer
123
You entered integer 123
Enter a float
2.35
You entered float 2.35

E:\Java>
```

Java program to find odd or even

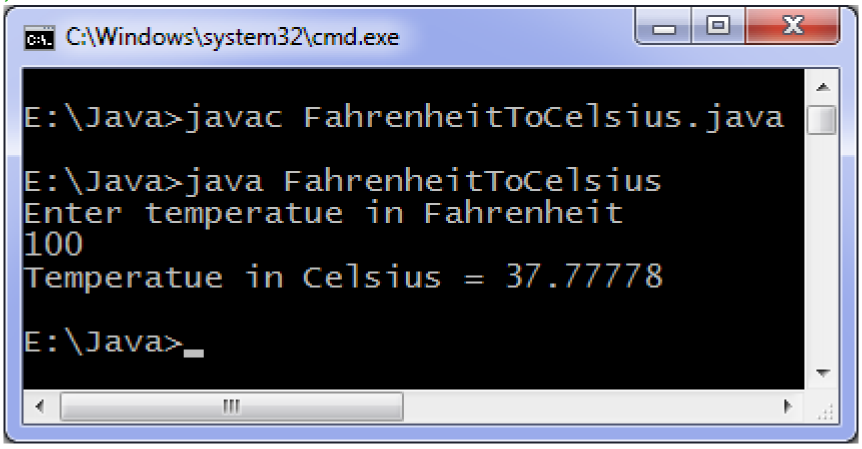
```
import java.util.Scanner;

class OddOrEven
{
   public static void main(String args[])
   {
      int x;
      System.out.println("Enter an integer to check if it is odd or even
");
      Scanner in = new Scanner(System.in);
      x = in.nextInt();

      if ( x % 2 == 0 )
         System.out.println("You entered an even number.");
      else
         System.out.println("You entered an odd number.");
   }
}
```



```
E:\Java>javac OddOrEven.java

E:\Java>java OddOrEven
Enter an integer to check if it is odd or even
6
You entered an even number.

E:\Java>java OddOrEven
Enter an integer to check if it is odd or even
5
You entered an odd number.

E:\Java>_
```

Java program to convert Fahrenheit to Celsius

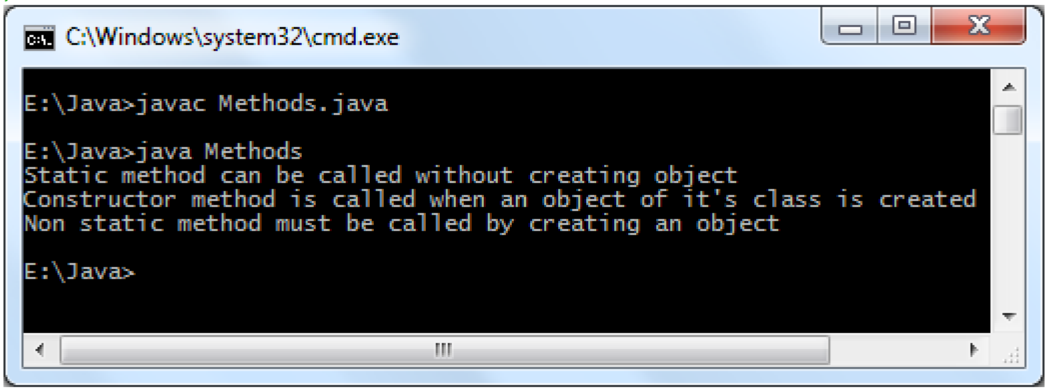
```java
import java.util.*;

class FahrenheitToCelsius {
  public static void main(String[] args) {
    float temperatue;
    Scanner in = new Scanner(System.in);

    System.out.println("Enter temperatue in Fahrenheit");
    temperatue = in.nextInt();

    temperatue = ((temperatue - 32)*5)/9;

    System.out.println("Temperatue in Celsius = " + temperatue);
  }
}
```



```
E:\Java>javac FahrenheitToCelsius.java

E:\Java>java FahrenheitToCelsius
Enter temperatue in Fahrenheit
100
Temperatue in Celsius = 37.77778

E:\Java>_
```

Java methods

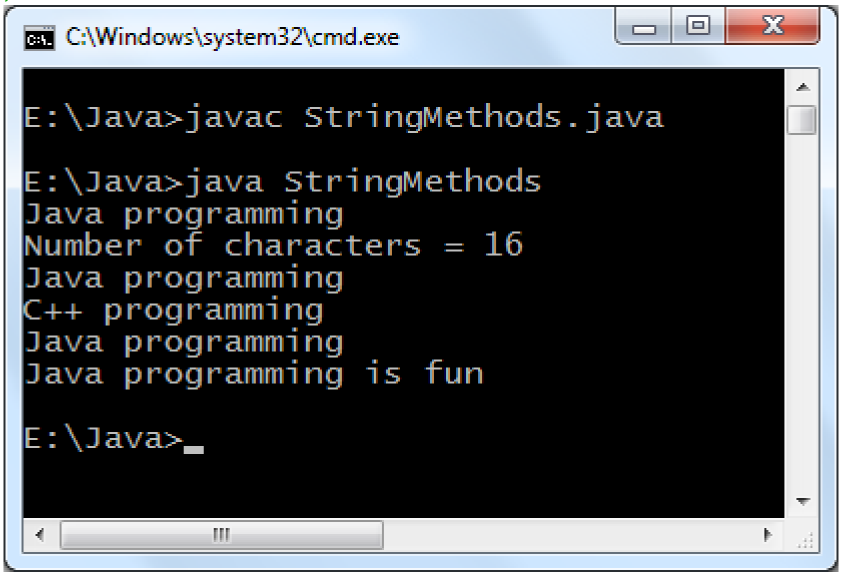
```
class Methods {

  // Constructor method

  Methods() {
    System.out.println("Constructor method is called when an object of it's
class is created");
  }

  // Main method where program execution begins

  public static void main(String[] args) {
    staticMethod();
    Methods object = new Methods();
    object.nonStaticMethod();
  }

  // Static method

  static void staticMethod() {
    System.out.println("Static method can be called without creating
object");
  }

  // Non static method

  void nonStaticMethod() {
    System.out.println("Non static method must be called by creating an
object");
  }
}
```



```
E:\Java>javac Methods.java

E:\Java>java Methods
Static method can be called without creating object
Constructor method is called when an object of it's class is created
Non static method must be called by creating an object

E:\Java>
```

Java String methods

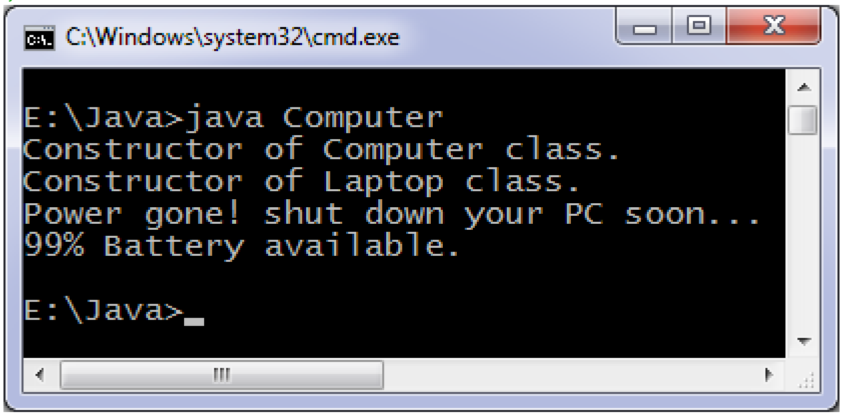
```java
class StringMethods
{
  public static void main(String args[])
  {
    int n;
    String s = "Java programming", t = "", u = "";

    System.out.println(s);

    // Find length of string

    n = s.length();
    System.out.println("Number of characters = " + n);

    // Replace characters in string

    t = s.replace("Java", "C++");
    System.out.println(s);
    System.out.println(t);

    // Concatenating string with another string

    u = s.concat(" is fun");
    System.out.println(s);
    System.out.println(u);
  }
}
```


C:\Windows\system32\cmd.exe

```
E:\Java>javac StringMethods.java

E:\Java>java StringMethods
Java programming
Number of characters = 16
Java programming
C++ programming
Java programming
Java programming is fun

E:\Java>_
```

Using two classes in Java program

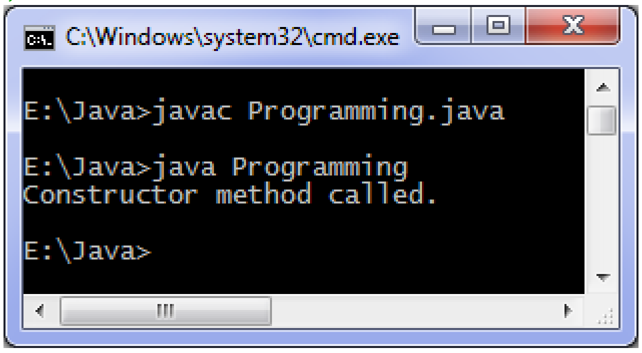
```java
class Computer {
  Computer() {
    System.out.println("Constructor of Computer class.");
  }

  void computer_method() {
    System.out.println("Power gone! Shut down your PC soon...");
  }

  public static void main(String[] args) {
    Computer my = new Computer();
    Laptop your = new Laptop();

    my.computer_method();
    your.laptop_method();
  }
}

class Laptop {
  Laptop() {
    System.out.println("Constructor of Laptop class.");
  }

  void laptop_method() {
    System.out.println("99% Battery available.");
  }
}
```



```
E:\Java>java Computer
Constructor of Computer class.
Constructor of Laptop class.
Power gone! shut down your PC soon...
99% Battery available.

E:\Java>_
```

Java constructor example

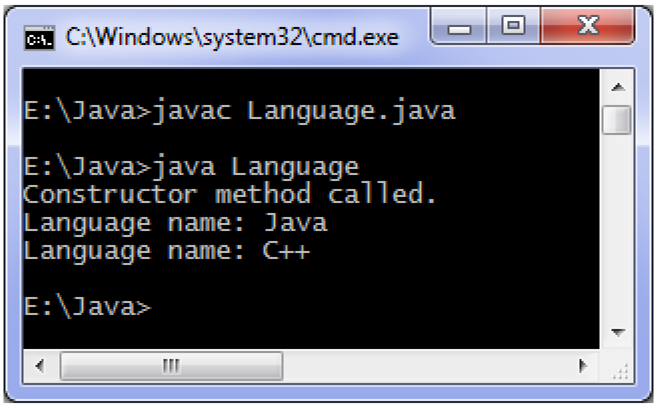
```java
class Programming {
  //constructor method
  Programming() {
    System.out.println("Constructor method called.");
  }

  public static void main(String[] args) {
    Programming object = new Programming(); //creating object
  }
}
```



```
E:\Java>javac Programming.java

E:\Java>java Programming
Constructor method called.

E:\Java>
```

Java constructor overloading

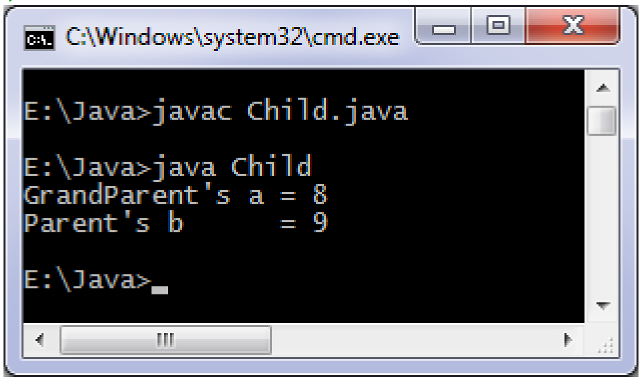
```java
class Language {
  String name;

  Language() {
    System.out.println("Constructor method called.");
  }

  Language(String t) {
    name = t;
  }

  public static void main(String[] args) {
    Language cpp  = new Language();
    Language java = new Language("Java");

    cpp.setName("C++");

    java.getName();
    cpp.getName();
  }

  void setName(String t) {
    name = t;
  }

  void getName() {
    System.out.println("Language name: " + name);
  }
}
```



```
E:\Java>javac Language.java

E:\Java>java Language
Constructor method called.
Language name: Java
Language name: C++

E:\Java>
```

Java constructor chaining

```java
class GrandParent {
  int a;

  GrandParent(int a) {
    this.a = a;
  }
}

class Parent extends GrandParent {
  int b;

  Parent(int a, int b) {
    super(a);
    this.b = b;
  }

  void show() {
    System.out.println("GrandParent's a = " + a);
    System.out.println("Parent's b      = " + b);
  }
}

class Child {
  public static void main(String[] args) {
    Parent object = new Parent(8, 9);
    object.show();
  }
}
```



```
E:\Java>javac Child.java

E:\Java>java Child
GrandParent's a = 8
Parent's b      = 9

E:\Java>_
```

Java program to swap two numbers

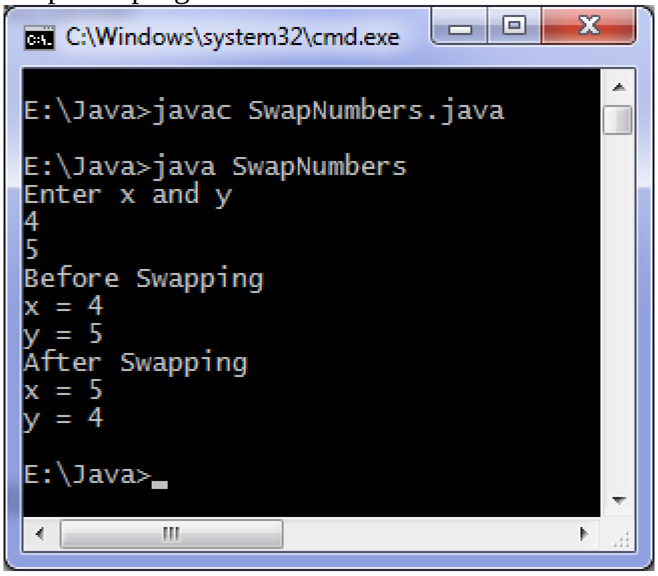
This java program swaps two numbers using a temporary variable. To swap numbers without using extra variable see another code below.

Swapping using temporary or third variable

```java
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\nx = "+x+"\ny = "+y);

      temp = x;
      x = y;
      y = temp;

      System.out.println("After Swapping\nx = "+x+"\ny = "+y);
   }
}
```

Swap numbers program class file.

Output of program:



```
E:\Java>javac SwapNumbers.java

E:\Java>java SwapNumbers
Enter x and y
4
5
Before Swapping
x = 4
y = 5
After Swapping
x = 5
y = 4

E:\Java>_
```

Swapping without temporary variable

```java
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\nx = "+x+"\ny = "+y);

      x = x + y;
      y = x - y;
      x = x - y;

      System.out.println("After Swapping\nx = "+x+"\ny = "+y);
   }
}
```

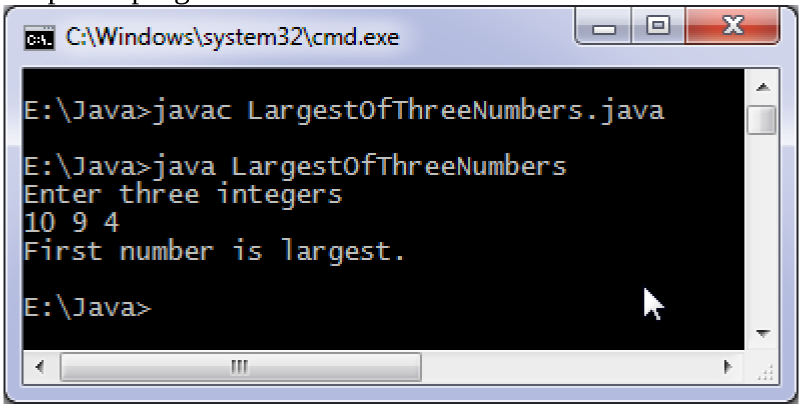
Java program to find largest of three numbers

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 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:



```
E:\Java>javac LargestOfThreeNumbers.java

E:\Java>java LargestOfThreeNumbers
Enter three integers
10 9 4
First number is largest.

E:\Java>
```

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:

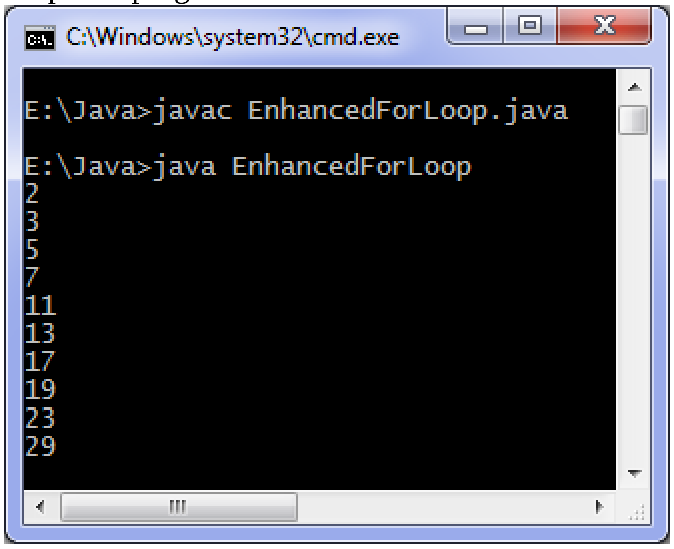
```
for (data_type variable: array_name)
```

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:



```
E:\Java>javac EnhancedForLoop.java

E:\Java>java EnhancedForLoop
2
3
5
7
11
13
17
19
23
29
```

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);
    }
  }
}
```

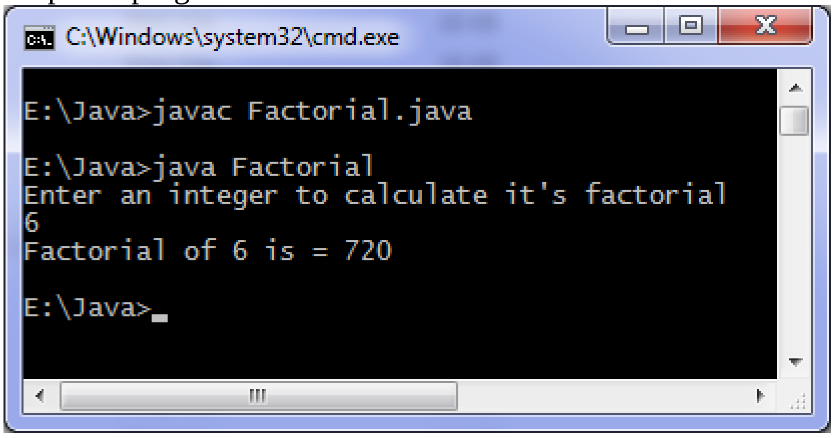
Java program to find factorial

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:



```
E:\Java>javac Factorial.java

E:\Java>java Factorial
Enter an integer to calculate it's factorial
6
Factorial of 6 is = 720

E:\Java>_
```

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! =
933262154439441526816992388562667004907159682643816214685929638952175999932
299156089414639761565182862536979208272237582511852109168640000000000000000
00000000
```

Java program print prime numbers

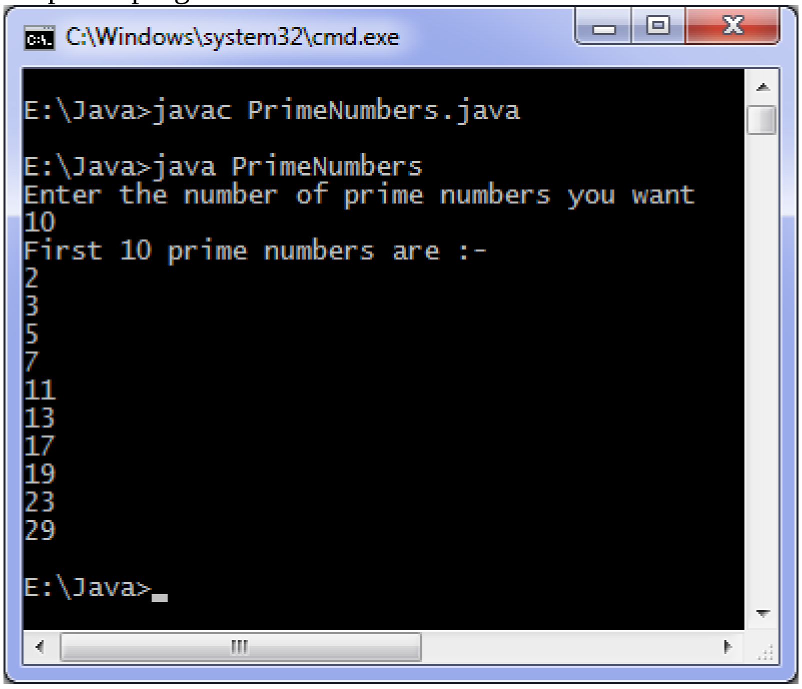
This java program prints prime numbers, number of prime numbers required is asked from the user. Remember that smallest prime number is 2.

Java programming code

```java
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 program class file.

Output of program:



```
E:\Java>javac PrimeNumbers.java

E:\Java>java PrimeNumbers
Enter the number of prime numbers you want
10
First 10 prime numbers are :-
2
3
5
7
11
13
17
19
23
29

E:\Java>_
```

Java program to reverse a string

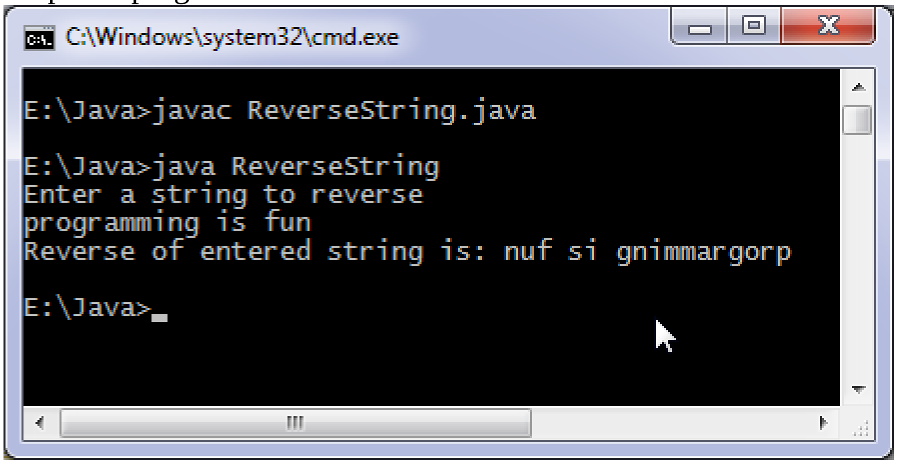
This java program reverses a string entered by the user. We use charAt method to extract characters from the string and append them in reverse order to reverse the entered string.

Java programming 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 program class file.

Output of program:



```
E:\Java>javac ReverseString.java

E:\Java>java ReverseString
Enter a string to reverse
programming is fun
Reverse of entered string is: nuf si gnimmargorp

E:\Java>_
```

Reverse string using StringBuffer class

```
class InvertString
{
   public static void main(String args[])
   {
      StringBuffer a = new StringBuffer("Java programming is 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".

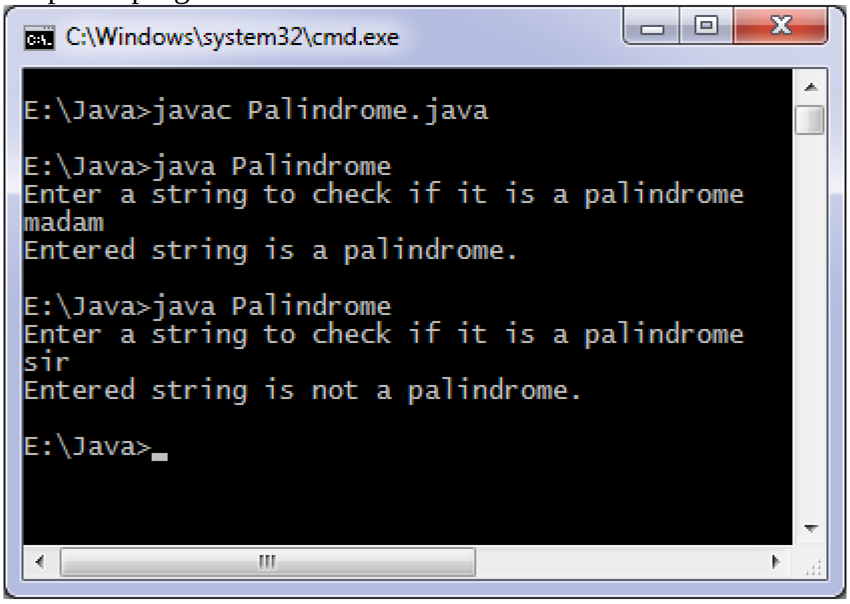
Java programming source code

```
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 program class file.

Output of program:



C:\Windows\system32\cmd.exe

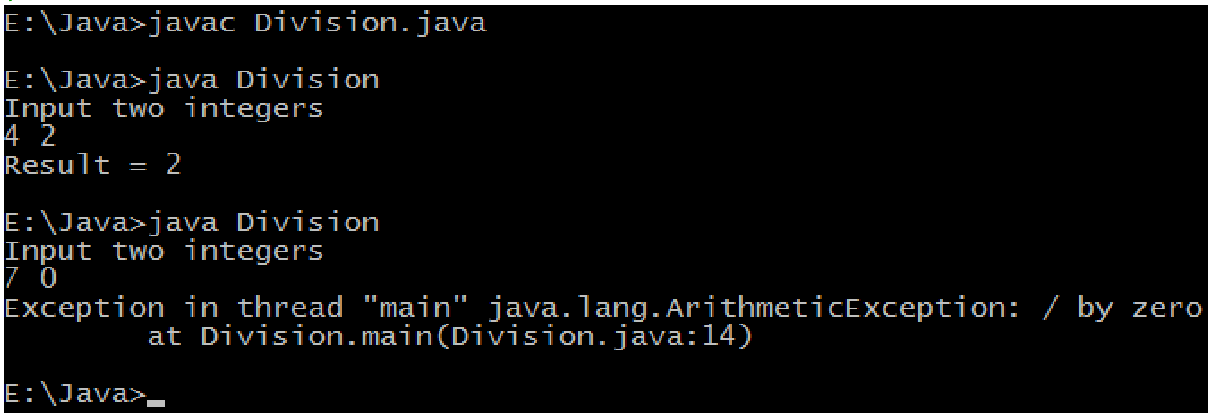
```
E:\Java>javac Palindrome.java

E:\Java>java Palindrome
Enter a string to check if it is a palindrome
madam
Entered string is a palindrome.

E:\Java>java Palindrome
Enter a string to check if it is a palindrome
sir
Entered string is not a palindrome.

E:\Java>_
```

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");
    }
  }
}
```

Java exception handling

```java
import java.util.Scanner;

class Division {
  public static void main(String[] args) {

  int a, b, result;

  Scanner input = new Scanner(System.in);
  System.out.println("Input two integers");

  a = input.nextInt();
  b = input.nextInt();

  result = a / b;

  System.out.println("Result = " + result);
  }
}
```



```
E:\Java>javac Division.java

E:\Java>java Division
Input two integers
4 2
Result = 2

E:\Java>java Division
Input two integers
7 0
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Division.main(Division.java:14)

E:\Java>_
```

Java exception handling example

```java
class Division {
  public static void main(String[] args) {

  int a, b, result;

  Scanner input = new Scanner(System.in);
  System.out.println("Input two integers");

  a = input.nextInt();
  b = input.nextInt();

  // try block

  try {
    result  = a / b;
    System.out.println("Result = " + result);
  }

  // catch block

  catch (ArithmeticException e) {
    System.out.println("Exception caught: Division by zero.");
    }
  }
}
```

Whenever an exception is caught corresponding catch block is executed, For example above code catches ArithmeticException only. If some other kind of exception is thrown it will not be caught so it's the programmer work to take care of all exceptions as in our try block we are performing arithmetic so we are capturing only arithmetic exceptions. A simple way to capture any exception is to use an object of Exception class as other classes inherit Exception class, see another example below:

```
class Exceptions {
  public static void main(String[] args) {

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

  try {
    for (int c = 1; c <= 5; c++) {
      System.out.println(languages[c]);
    }
  }
  catch (Exception e) {
    System.out.println(e);
  }
  }
}
```

Output of program:

```
C++
Java
Perl
Python
java.lang.ArrayIndexOutOfBoundsException: 5
```

Finally block in Java

Finally block is always executed whether an exception is thrown or not.

```
class Allocate {
  public static void main(String[] args) {

    try {
      long data[] = new long[1000000000];
    }
    catch (Exception e) {
      System.out.println(e);
    }

    finally {
      System.out.println("finally block will execute always.");
    }
  }
}
```

Output of program:

```
finally block will execute always.
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
at Allocate.main(Allocate.java:5)
```

Fibonacci series

```
package com.java2novice.algos;

public class MyFibonacci {

    public static void main(String a[]){

         int febCount = 15;
         int[] feb = new int[febCount];
         feb[0] = 0;
         feb[1] = 1;
         for(int i=2; i < febCount; i++){
             feb[i] = feb[i-1] + feb[i-2];
         }

         for(int i=0; i< febCount; i++){
                 System.out.print(feb[i] + " ");
         }
    }
}
```

Output:

```
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

Java program for linear search – Example

Example Program:

This program uses linear search algorithm to find out a number among all other numbers entered by user.

```
/* Program: Linear Search Example
 * Input: Number of elements, element's values, value to be searched
 * Output:Position of the number input by user among other numbers*/
import java.util.Scanner;
class LinearSearchExample
{
   public static void main(String args[])
   {
      int counter, num, item, array[];
      //To capture user input
      Scanner input = new Scanner(System.in);
      System.out.println("Enter number of elements:");
      num = input.nextInt();
      //Creating array to store the all the numbers
      array = new int[num];
      System.out.println("Enter " + num + " integers");
      //Loop to store each numbers in array
      for (counter = 0; counter < num; counter++)
        array[counter] = input.nextInt();

      System.out.println("Enter the search value:");
      item = input.nextInt();

      for (counter = 0; counter < num; counter++)
      {
         if (array[counter] == item)
         {
           System.out.println(item+" is present at location "+(counter+1));
           /*Item is found so to stop the search and to come out of the
            * loop use break statement.*/
           break;
         }
      }
      if (counter == num)
        System.out.println(item + " doesn't exist in array.");
   }
}
```

Output 1:

```
Enter number of elements:
6
Enter 6 integers
22
33
45
1
3
99
Enter the search value:
45
```

```
45 is present at location 3
```

Output 2:

```
Enter number of elements:
4
Enter 4 integers
11
22
4
5
Enter the search value:
99
99 doesn't exist in array.
```

Java program to perform binary search – Example

Example Program to perform binary search on a list of integer numbers

This program uses binary search algorithm to search an element in given list of elements.

```
/* Program: Binary Search Example
 * Input: Number of elements, element's values, value to be searched
 * Output:Position of the number input by user among other numbers*/
import java.util.Scanner;
class BinarySearchExample
{
   public static void main(String args[])
   {
      int counter, num, item, array[], first, last, middle;
      //To capture user input
      Scanner input = new Scanner(System.in);
      System.out.println("Enter number of elements:");
      num = input.nextInt();

      //Creating array to store the all the numbers
      array = new int[num];

      System.out.println("Enter " + num + " integers");
      //Loop to store each numbers in array
      for (counter = 0; counter < num; counter++)
          array[counter] = input.nextInt();

      System.out.println("Enter the search value:");
      item = input.nextInt();
      first = 0;
      last = num - 1;
      middle = (first + last)/2;

      while( first <= last )
      {
         if ( array[middle] < item )
           first = middle + 1;
         else if ( array[middle] == item )
         {
           System.out.println(item + " found at location " + (middle + 1) +
".");
           break;
         }
         else
         {
             last = middle - 1;
         }
         middle = (first + last)/2;
      }
      if ( first > last )
          System.out.println(item + " is not found.\n");
   }
}
```

Output 1:

```
Enter number of elements:
7
Enter 7 integers
4
5
66
77
8
99
0
Enter the search value:
77
77 found at location 4.
```

Output 2:

```
Enter number of elements:
5
Enter 5 integers
12
3
77
890
23
Enter the search value:
99
99 is not found.
```

Program 1: Reverse a number using while Loop

The program will prompt user to input the number and then it will reverse the same number using while loop.

```
import java.util.Scanner;
class ReverseNumberWhile
{
   public static void main(String args[])
   {
      int num=0;
      int reversenum =0;
      System.out.println("Input your number and press enter: ");
      //This statement will capture the user input
      Scanner in = new Scanner(System.in);
      //Captured input would be stored in number num
      num = in.nextInt();
      //While Loop: Logic to find out the reverse number
      while( num != 0 )
      {
          reversenum = reversenum * 10;
          reversenum = reversenum + num%10;
          num = num/10;
      }

      System.out.println("Reverse of input number is: "+reversenum);
   }
}
```

Output:

```
Input your number and press enter:
145689
Reverse of input number is: 986541
```

Program 2: Reverse a number using for Loop

```
import java.util.Scanner;
class ForLoopReverseDemo
{
   public static void main(String args[])
   {
      int num=0;
      int reversenum =0;
      System.out.println("Input your number and press enter: ");
      //This statement will capture the user input
      Scanner in = new Scanner(System.in);
      //Captured input would be stored in number num
      num = in.nextInt();
      /* for loop: No initialization part as num is already
       * initialized and no increment/decrement part as logic
       * num = num/10 already decrements the value of num
       */
      for( ;num != 0; )
      {
          reversenum = reversenum * 10;
          reversenum = reversenum + num%10;
          num = num/10;
      }

      System.out.println("Reverse of specified number is: "+reversenum);
   }
}
```

Output:

```
Input your number and press enter:
56789111
Reverse of specified number is: 11198765
```

Program 3: Reverse a number using recursion

```
import java.util.Scanner;
class RecursionReverseDemo
{
   //A method for reverse
   public static void reverseMethod(int number) {
       if (number < 10) {
           System.out.println(number);
           return;
       }
       else {
           System.out.print(number % 10);
           //Method is calling itself: recursion
           reverseMethod(number/10);
       }
   }
   public static void main(String args[])
   {
        int num=0;
        System.out.println("Input your number and press enter: ");
        Scanner in = new Scanner(System.in);
        num = in.nextInt();
        System.out.print("Reverse of the input number is:");
        reverseMethod(num);
        System.out.println();
   }
}
```

Output:

```
Input your number and press enter:
5678901
Reverse of the input number is:1098765
```

Example: Reverse an already initialized number
In all the above programs we are prompting user for the input number, however if do not want the user interaction part and want to reverse an initialized number then this is how you can do it.

```
class ReverseNumberDemo
{
   public static void main(String args[])
   {
      int num=123456789;
      int reversenum =0;
      while( num != 0 )
      {
          reversenum = reversenum * 10;
          reversenum = reversenum + num%10;
          num = num/10;
      }

      System.out.println("Reverse of specified number is: "+reversenum);
   }
}
```

Output:

```
Reverse of specified number is: 987654321
```