**1. Java simple calculator**

1. A four-function calculator with the following functions:  
\* Addition – adds two numbers: n1 + n2.  
\* Subtraction – subtracts number two from number one: n1 – n2.  
\* Multiplication – Multiplies two numbers: n1 \* n2.  
\* Division – divides number two into number one: n1 / n2.

2. Used JButton for the numbers and functions.

3. Used Jmenu for File and help.  
 4. Created a class to respond to the events caused by numbers, functions, exit, help and about functionality.

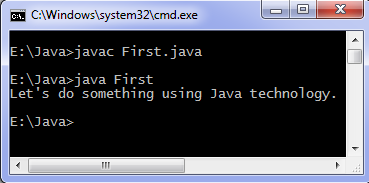
5. Used **BorderLayout** to layout various components in the Calculator Frame. A BorderLayout lays out a container, arranging  
and resizing its components to fit in five regions: north, south, east, west, and center.

2. Example 1: Display message on computer screen.

class First {

public static void main(String[] arguments) {

System.out.println("Let's do something using Java technology."); }}



Example 2: Print integers

class Integers {

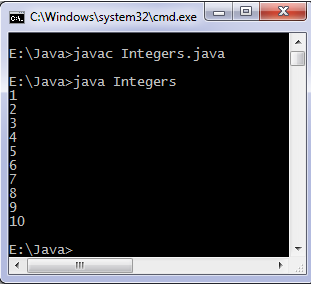
public static void main(String[] arguments)

int c; //declaring a variable

/\* Using for loop to repeat instruction execution \*/

for (c = 1; c <= 10; c++) {

System.out.println(c); } }}



Example: 3 If else control instructions:

class Condition {

public static void main(String[] args) {

boolean learning = true;

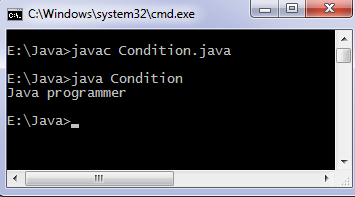
if (learning) {

System.out.println("Java programmer");

} else {

System.out.println("What are you doing here?");

} }}



Example: 4 Command line arguments:

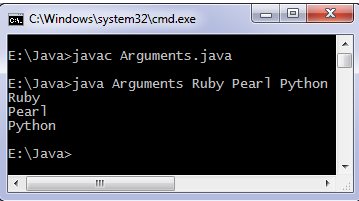
class Arguments {

public static void main(String[] args) {

for (String t: args) {

System.out.println(t);

} }}

****

**For loop example:**

**classForLoopExample {**

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

**for(int i=10; i>1; i--){**

**System.out.println("The value of i is: "+i);**

**}**

**}**

**}**

**The output of this program is:**

**The value of i is: 10**

**The value of i is: 9**

**The value of i is: 8**

**The value of i is: 7**

**The value of i is: 6**

**The value of i is: 5**

**The value of i is: 4**

**The value of i is: 3**

**The value of i is: 2**

Infinite for loop: The importance of Boolean expression and increment/decrement operation co-ordination:

class ForLoopExample2 {

public static void main(String args[]){

for(int i=1; i>=1; i++){

System.out.println("The value of i is: "+i);

} }}

Here is another example of infinite for loop:

// infinite loop

for ( ; ; ) {

// statement(s)

}

For loop example to iterate an array:

Here we are iterating and displaying array elements using the for loop.

class ForLoopExample3 {

public static void main(String args[]){

intarr[4]={2,11,45,9};

//i starts with 0 as array index starts with 0 too

for(int i=0; i<4; i++){

System.out.println(arr[i]);

} } }

Output:

2

11

45

9

Java program to sum the elements of an array

Program 1: No user interaction

classSumOfArray{

public static void main(String args[]){

int[] array = {10, 20, 30, 40, 50, 10};

int sum = 0;

//Advanced for loop

for(intnum : array) {

sum = sum+num;

}System.out.println("Sum of array elements is:"+sum);

}}

Output:

Sum of array elements is:160

Program 2: User enters the array’s elements

importjava.util.Scanner;

classSumDemo{

public static void main(String args[]){

Scanner scanner = new Scanner(System.in);

int[] array = new int[10];

int sum = 0;

System.out.println("Enter the elements:");

for (int i=0; i<10; i++)

{

array[i] = scanner.nextInt();

}

for(intnum : array) {

sum = sum+num;

}

System.out.println("Sum of array elements is:"+sum);

}}

Output:

Enter the elements:

1

2

3

4

5

6

7

8

9

10

Sum of array elements is:55

Java Program to Calculate Area of Rectangle

importjava.util.Scanner;

classAreaOfRectangle {

public static void main (String[] args)

{

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the length of Rectangle:");

double length = scanner.nextDouble();

System.out.println("Enter the width of Rectangle:");

double width = scanner.nextDouble();

//Area = length\*width;

double area = length\*width;

System.out.println("Area of Rectangle is:"+area);

}

}

Output:

Enter the length of Rectangle:

2

Enter the width of Rectangle:

8

Area of Rectangle is:16.0

Program 2:

class AreaOfRectangle2 {

public static void main (String[] args)

{

double length = 4.5;

double width = 8.0;

double area = length\*width;

System.out.println("Area of Rectangle is:"+area);

}

}

Output:

Area of Rectangle is:36.0

Java program to display prime numbers from 1 to 100 and 1 to n

Program to display the prime numbers from 1 to 100

It will display the prime numbers between 1 and 100.

classPrimeNumbers

{

public static void main (String[] args)

{

int i =0;

intnum =0;

//Empty String

String primeNumbers = "";

for (i = 1; i <= 100; i++)

{

int counter=0;

for(num =i; num>=1; num--)

{

if(i%num==0)

{

counter = counter + 1;

}

}

if (counter ==2)

{

//Appended the Prime number to the String

primeNumbers = primeNumbers + i + " ";

}

}

System.out.println("Prime numbers from 1 to 100 are :");

System.out.println(primeNumbers);

}

}

Output:

Prime numbers from 1 to 100 are :

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Program to display prime numbers from 1 to n

It will display all the prime numbers between 1 and n (the number entered by user).

importjava.util.Scanner;

class PrimeNumbers2

{

public static void main (String[] args)

{

Scanner scanner = new Scanner(System.in);

int i =0;

intnum =0;

//Empty String

String primeNumbers = "";

System.out.println("Enter the value of n:");

int n = scanner.nextInt();

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

{

int counter=0;

for(num =i; num>=1; num--)

{

if(i%num==0)

{

counter = counter + 1;

}

} if (counter ==2){

//Appended the Prime number to the String

primeNumbers = primeNumbers + i + " ";

} }

System.out.println("Prime numbers from 1 to n are :");

System.out.println(primeNumbers);

}}

Output:

Enter the value of n:

150

Prime numbers from 1 to n are :

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101

Java program for binary to decimal conversion

There are two following ways to convert binary number to decimal number:

1) Using Integer.parseInt() method of Integer class.

2) Do conversion by writing your own logic without using any predefined methods.

Method 1: Binary to Decimal conversion using Integer.parseInt() method

importjava.util.Scanner;

classBinaryToDecimal {

public static void main(String args[]){

Scanner input = new Scanner( System.in );

System.out.print("Enter a binary number: ");

String binaryString =input.nextLine();

System.out.println("Output: "+Integer.parseInt(binaryString,2));

}}

Output:

Enter a binary number: 1101

Output: 13

Method 2: Conversion without using parseInt

public class Details {

publicintBinaryToDecimal(intbinaryNumber){

int decimal = 0;

int p = 0;

while(true){

if(binaryNumber == 0){

break;

} else {

int temp = binaryNumber%10;

decimal += temp\*Math.pow(2, p);

binaryNumber = binaryNumber/10;

p++; } }

return decimal;

}

public static void main(String args[]){

Details obj = new Details();

System.out.println("110 --> "+obj.BinaryToDecimal(110));

System.out.println("1101 --> "+obj.BinaryToDecimal(1101));

System.out.println("100 --> "+obj.BinaryToDecimal(100));

System.out.println("110111 --> "+obj.BinaryToDecimal(110111));

}}

Output:

110 --> 6

1101 --> 13

100 --> 4

110111 --> 55

Java program to check prime number

This program will prompt user to enter a number and then it will check and display whether the input number is prime or not.

importjava.util.Scanner;

classPrimeCheck

{

public static void main(String args[])

{

int temp;

booleanisPrime=true;

Scanner scan= new Scanner(System.in);

System.out.println("Enter a number for check:");

//capture the input in an integer

intnum=scan.nextInt();

for(int i=2;i<=num/2;i++)

{

temp=num%i;

if(temp==0)

{

isPrime=false;

break;

}}

//If isPrime is true then the number is prime else not

if(isPrime)

System.out.println(num + " is Prime Number");

else

System.out.println(num + " is not Prime Number");

}}

Output:

Enter a number for check:

19

19 is Prime Number

Output 2:

Enter a number for check:

6

6 is not Prime Number

Java program for bubble sort in Ascending & descending order

Bubble sort program for sorting in ascending Order

importjava.util.Scanner;

classBubbleSortExample {

public static void main(String []args) {

intnum, i, j, temp;

Scanner input = new Scanner(System.in);

System.out.println("Enter the number of integers to sort:");

num = input.nextInt();

int array[] = new int[num];

System.out.println("Enter " + num + " integers: ");

for (i = 0; i <num; i++)

array[i] = input.nextInt();

for (i = 0; i < ( num - 1 ); i++) {

for (j = 0; j <num - i - 1; j++) {

if (array[j] > array[j+1])

{ temp = array[j];

array[j] = array[j+1];

array[j+1] = temp;

} } }

System.out.println("Sorted list of integers:");

for (i = 0; i <num; i++)

System.out.println(array[i]);

}}

Output:

Enter the number of integers to sort:

6

Enter 6 integers:

12

6

78

9

45

08

Sorted list of integers:

6

8

9

12

45

78

Bubble sort program for sorting in descending Order

In order to sort in descending order we just need to change the logic array[j] >array[j+1] to array[j] < array[j+1] in the above program. Complete code as follows:

importjava.util.Scanner;

classBubbleSortExample {

public static void main(String []args) {

intnum, i, j, temp;

Scanner input = new Scanner(System.in);

System.out.println("Enter the number of integers to sort:");

num = input.nextInt();

int array[] = new int[num];

System.out.println("Enter " + num + " integers: ");

for (i = 0; i <num; i++)

array[i] = input.nextInt();

for (i = 0; i < ( num - 1 ); i++) {

for (j = 0; j <num - i - 1; j++) {

if (array[j] < array[j+1])

{ temp = array[j];

array[j] = array[j+1];

array[j+1] = temp;

} } }

System.out.println("Sorted list of integers:");

for (i = 0; i <num; i++)

System.out.println(array[i]);

} }

Output:

Enter the number of integers to sort:

6

Enter 6 integers:

89

12

45

9

56

102

Sorted list of integers:

102

89

56

45

12

9

Java Program to check Even or Odd number

importjava.util.Scanner;

classCheckEvenOdd

{

public static void main(String args[])

{

intnum;

System.out.println("Enter an Integer number:");

//The input provided by user is stored in num

Scanner input = new Scanner(System.in);

num = input.nextInt();

/\* If number is divisible by 2 then it's an even number

\* else odd number\*/

if ( num % 2 == 0 )

System.out.println("Entered number is even");

else

System.out.println("Entered number is odd");

}}

Output 1:

Enter an Integer number:

78

Entered number is even

Output 2:

Enter an Integer number:

77

Entered number is odd

How to convert a char array to a string in Java?

There are two ways to convert a char array (char[]) to String in Java:

1) Creating String object by passing array name to the constructor

2) Using valueOf() method of String class.

Example:

This example demonstrates both the above mentioned ways of converting a char array to String. Here we have a char array ch and we have created two strings str and str1 using the char array.

classCharArrayToString

{ public static void main(String args[])

{ // Method 1: Using String object

char[] ch = {'g', 'o', 'o', 'd', ' ', 'm', 'o', 'r', 'n', 'i', 'n', 'g'};

String str = new String(ch);

System.out.println(str);

// Method 2: Using valueOf method

String str2 = String.valueOf(ch);

System.out.println(str2);

}}

Output:

good morning

good morning

Java program to get IP address

In this example we are gonna see how to get IP address of a System. The steps are as follows:

1) Get the local host address by calling getLocalHost() method of InetAddress class.

2) Get the IP address by calling getHostAddress() method.

importjava.net.InetAddress;

classGetMyIPAddress

{

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

/\* public static InetAddressgetLocalHost()

\* throws UnknownHostException: Returns the address

\* of the local host. This is achieved by retrieving

\* the name of the host from the system, then resolving

\* that name into an InetAddress. Note: The resolved

\* address may be cached for a short period of time.

\*/

InetAddressmyIP=InetAddress.getLocalHost();

/\* public String getHostAddress(): Returns the IP

\* address string in textual presentation.

\*/

System.out.println("My IP Address is:");

System.out.println(myIP.getHostAddress());

}

}

Output:

My IP Address is:

115.242.7.243

How to open notepad using Java program

importjava.util.\*;

import java.io.\*;

class Notepad {

public static void main(String[] args) {

Runtime rs = Runtime.getRuntime();

try {

rs.exec("notepad");

}

catch (IOException e) {

System.out.println(e);

}

}

}

Java programming code

importjava.util.\*;

classGetCurrentDateAndTime

{

public static void main(String args[])

{

int day, month, year;

int second, minute, hour;

GregorianCalendar date = new GregorianCalendar();

day = date.get(Calendar.DAY\_OF\_MONTH);

month = date.get(Calendar.MONTH);

year = date.get(Calendar.YEAR);

second = date.get(Calendar.SECOND);

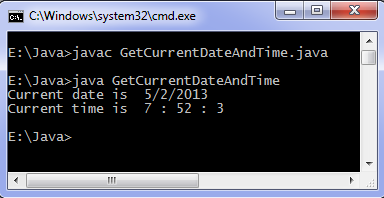
minute = date.get(Calendar.MINUTE);

hour = date.get(Calendar.HOUR);

System.out.println("Current date is "+day+"/"+(month+1)+"/"+year);

System.out.println("Current time is "+hour+" : "+minute+" : "+second);

}

}

Swap Numbers Java Example

/\*

Swap Numbers Java Example

This Swap Numbers Java Example shows how to

swap value of two numbers using java.

\*/

public class SwapElementsExample {

public static void main(String[] args) {

int num1 = 10;

int num2 = 20;

System.out.println("Before Swapping");

System.out.println("Value of num1 is :" + num1);

System.out.println("Value of num2 is :" +num2);

//swap the value

swap(num1, num2);

}

private static void swap(int num1, int num2) {

int temp = num1;

num1 = num2;

num2 = temp;

System.out.println("After Swapping");

System.out.println("Value of num1 is :" + num1);

System.out.println("Value of num2 is :" +num2);

}

}

/\*

Output of Swap Numbers example would be

Before Swapping

Value of num1 is :10

Value of num2 is :20

After Swapping

Value of num1 is :20

Value of num2 is :10

\*/

Swap Numbers Without Using Third Variable Java Example

/\*

Swap Numbers Without Using Third Variable Java Example

This Swap Numbers Java Example shows how to

swap value of two numbers without using third variable using java.

\*/

public class SwapElementsWithoutThirdVariableExample {

public static void main(String[] args) {

int num1 = 10;

int num2 = 20;

System.out.println("Before Swapping");

System.out.println("Value of num1 is :" + num1);

System.out.println("Value of num2 is :" +num2);

//add both the numbers and assign it to first

num1 = num1 + num2;

num2 = num1 - num2;

num1 = num1 - num2;

System.out.println("Before Swapping");

System.out.println("Value of num1 is :" + num1);

System.out.println("Value of num2 is :" +num2);

}

}

/\*

Output of Swap Numbers Without Using Third Variable example would be

Before Swapping

Value of num1 is :10

Value of num2 is :20

Before Swapping

Value of num1 is :20

Value of num2 is :10

\*/

Find Largest and Smallest Number in an Array Example

/\*

Find Largest and Smallest Number in an Array Example

This Java Example shows how to find largest and smallest number in an

array.

\*/

public class FindLargestSmallestNumber {

public static void main(String[] args) {

//array of 10 numbers

int numbers[] = new int[]{32,43,53,54,32,65,63,98,43,23};

//assign first element of an array to largest and smallest

int smallest = numbers[0];

intlargetst = numbers[0];

for(int i=1; i<numbers.length; i++)

{

if(numbers[i] >largetst)

largetst = numbers[i];

else if (numbers[i] < smallest)

smallest = numbers[i];

}

System.out.println("Largest Number is : " + largetst);

System.out.println("Smallest Number is : " + smallest);

}

}

/\*

Output of this program would be

Largest Number is : 98

Smallest Number is : 23

\*/

**Calculate Circle Area using Java Example**

1. */\**
2. *Calculate Circle Area using Java Example*
3. *This Calculate Circle Area using Java Example shows how to calculate*
4. *area of circle using it's radius.*
5. *\*/*
7. **import** java.io.BufferedReader;
8. **import** java.io.IOException;
9. **import** java.io.InputStreamReader;
11. **public** **class** CalculateCircleAreaExample {
13. **public** **static** **void** main(String[] args) {
15. **int** radius = 0;
16. System.out.println("Please enter radius of a circle");
18. **try**
19. {
20. *//get the radius from console*
21. BufferedReader br = **new** BufferedReader(**new**InputStreamReader(System.in));
22. radius = Integer.parseInt(br.readLine());
23. }
24. *//if invalid value was entered*
25. **catch**(NumberFormatException ne)
26. {
27. System.out.println("Invalid radius value" + ne);
28. System.exit(0);
29. }
30. **catch**(IOException ioe)
31. {
32. System.out.println("IO Error :" + ioe);
33. System.exit(0);
34. }
36. */\**
37. *\* Area of a circle is*
38. *\* pi \* r \* r*
39. *\* where r is a radius of a circle.*
40. *\*/*
42. *//NOTE : use Math.PI constant to get value of pi*
43. **double** area = Math.PI \* radius \* radius;
45. System.out.println("Area of a circle is " + area);
46. }
47. }
49. */\**
50. *Output of Calculate Circle Area using Java Example would be*
51. *Please enter radius of a circle*
52. *19*
53. *Area of a circle is 1134.1149479459152*
54. *\*/*

**Draw Oval & Circle in Applet Window Example**

1. */\**
2. *Draw Oval & Circle in Applet Window Example*
3. *This java example shows how to draw ovals & circles in an applet window using*
4. *drawOval method of Graphics class. It also shows how to draw a filled*
5. *ovals and circles using fillOval method of Graphics class.*
6. *\*/*
8. */\**
9. *<applet code="DrawOvalsExample" width=500 height=500>*
10. *</applet>*
11. *\*/*

14. **import** java.applet.Applet;
15. **import** java.awt.Color;
16. **import** java.awt.Graphics;
18. **public** **class** DrawOvalsExample **extends** Applet{
20. **public** **void** paint(Graphics g){
22. *//set color to red*
23. setForeground(Color.red);
25. */\**
26. *\* to draw a oval in an applet window use,*
27. *\* void drawOval(int x1,int y1, int width, int height)*
28. *\* method.*
29. *\**
30. *\* This method draws a oval of specified width and*
31. *\* height at (x1,y1)*
32. *\*/*
34. *//this will draw a oval of width 50 & height 100 at (10,10)*
35. g.drawOval(10,10,50,100);

38. */\**
39. *\* To draw a filled oval use*
40. *\* fillOval(int x1,int y1, int width, int height)*
41. *\* method of Graphics class.*
42. *\*/*
44. *//draw filled oval*
45. g.fillOval(100,20,50,100);
47. }
48. }

# Read line of characters from console using InputStream

1. */\**
2. *Read line of characters from console using InputStream*
3. *This example shows how to read a line or string from console window*
4. *usingreadLine method of BufferedInputStream.*
5. *\*/*
7. **import** java.io.BufferedReader;
8. **import** java.io.IOException;
9. **import** java.io.InputStreamReader;
11. **public** **class** ReadLineFromConsoleExample {
13. **public** **static** **void** main(String[] args) {
15. */\**
16. *\* To read line or string from console use,*
17. *\* readLine method of BufferedReader class.*
18. *\*/*

21. BufferedReader br =
22. **new** BufferedReader(**new** InputStreamReader(System.in));
24. String strLine = **null**;
26. System.out.println("Reading line of characters from console");
27. System.out.println("Enter exit to quit");
29. **try**
30. {
32. **while**( (strLine = br.readLine()) != **null**)
33. {
34. **if**(strLine.equals("exit"))
35. **break**;
37. System.out.println("Line entered : "  + strLine);
39. }
41. br.close();
43. }
44. **catch**(Exception e)
45. {
46. System.out.println("Error while reading line from console : " + e);
47. }
48. }
49. }

[**5**](http://www.java-examples.com/read-line-characters-console-using-inputstream)

**Read Number from Console and Check if it is a Palindrome Number**

1. */\**
2. *Read Number from Console and Check if it is a Palindrome Number*
3. *This Java example shows how to input the number from console and*
4. *check if the number is a palindrome number or not.*
5. *\*/*
7. **import** java.io.BufferedReader;
8. **import** java.io.IOException;
9. **import** java.io.InputStreamReader;
11. **public** **class** InputPalindromeNumberExample {
13. **public** **static** **void** main(String[] args) {
15. System.out.println("Enter the number to check..");
16. **int** number = 0;
18. **try**
19. {
20. *//take input from console*
21. BufferedReader br = **new** BufferedReader(**new**InputStreamReader(System.in));
22. *//parse the line into int*
23. number = Integer.parseInt(br.readLine());
25. }
26. **catch**(NumberFormatException ne)
27. {
28. System.out.println("Invalid input: " + ne);
29. System.exit(0);
30. }
31. **catch**(IOException ioe)
32. {
33. System.out.println("I/O Error: " + ioe);
34. System.exit(0);
35. }
37. System.out.println("Number is " + number);
38. **int** n = number;
39. **int** reversedNumber  = 0;
40. **int** temp=0;
42. *//reverse the number*
43. **while**(n > 0){
44. temp = n % 10;
45. n = n / 10;
46. reversedNumber = reversedNumber \* 10 + temp;
47. }
49. */\**
50. *\* if the number and it's reversed number are same, the number is a*
51. *\* palindrome number*
52. *\*/*
53. **if**(number == reversedNumber)
54. System.out.println(number + " is a palindrome number");
55. **else**
56. System.out.println(number + " is not a palindrome number");
57. }
59. }
61. */\**
62. *Output of the program would be*
63. *Enter the number to check..*
64. *121*
65. *Number is 121*
66. *121 is a palindrome number*
67. *\*/*

**Calculate Circle Perimeter using Java Example**

1. */\**
2. *Calculate Circle Perimeter using Java Example*
3. *This Calculate Circle Perimeter using Java Example shows how to calculate*
4. *Perimeter of circle using it's radius.*
5. *\*/*
7. **import** java.io.BufferedReader;
8. **import** java.io.IOException;
9. **import** java.io.InputStreamReader;
11. **public** **class** CalculateCirclePerimeterExample {
13. **public** **static** **void** main(String[] args) {
15. **int** radius = 0;
16. System.out.println("Please enter radius of a circle");
18. **try**
19. {
20. *//get the radius from console*
21. BufferedReader br = **new** BufferedReader(**new**InputStreamReader(System.in));
22. radius = Integer.parseInt(br.readLine());
23. }
24. *//if invalid value was entered*
25. **catch**(NumberFormatException ne)
26. {
27. System.out.println("Invalid radius value" + ne);
28. System.exit(0);
29. }
30. **catch**(IOException ioe)
31. {
32. System.out.println("IO Error :" + ioe);
33. System.exit(0);
34. }
36. */\**
37. *\* Perimeter of a circle is*
38. *\* 2 \* pi \* r*
39. *\* where r is a radius of a circle.*
40. *\*/*
42. *//NOTE : use Math.PI constant to get value of pi*
43. **double** perimeter = 2 \* Math.PI \* radius;
45. System.out.println("Perimeter of a circle is " + perimeter);
46. }
47. }
49. */\**
50. *Output of Calculate Circle Perimeter using Java Example would be*
51. *Please enter radius of a circle*
52. *19*
53. *Perimeter of a circle is 119.38052083641213*
54. *\*/*

**Calculate Rectangle Area using Java Example**

1. */\**
2. *Calculate Rectangle Area using Java Example*
3. *This Calculate Rectangle Area using Java Example shows how to calculate*
4. *area of Rectangle using it's length and width.*
5. *\*/*
7. **import** java.io.BufferedReader;
8. **import** java.io.IOException;
9. **import** java.io.InputStreamReader;
11. **public** **class** CalculateRectArea {
13. **public** **static** **void** main(String[] args) {
15. **int** width = 0;
16. **int** length = 0;
18. **try**
19. {
20. *//read the length from console*
21. BufferedReader br = **new** BufferedReader(**new**InputStreamReader(System.in));
23. System.out.println("Please enter length of a rectangle");
24. length = Integer.parseInt(br.readLine());
26. *//read the width from console*
27. System.out.println("Please enter width of a rectangle");
28. width = Integer.parseInt(br.readLine());

31. }
32. *//if invalid value was entered*
33. **catch**(NumberFormatException ne)
34. {
35. System.out.println("Invalid value" + ne);
36. System.exit(0);
37. }
38. **catch**(IOException ioe)
39. {
40. System.out.println("IO Error :" + ioe);
41. System.exit(0);
42. }
44. */\**
45. *\* Area of a rectangle is*
46. *\* length \* width*
47. *\*/*
49. **int** area = length \* width;
51. System.out.println("Area of a rectangle is " + area);
52. }
54. }
56. */\**
57. *Output of Calculate Rectangle Area using Java Example would be*
58. *Please enter length of a rectangle*
59. *10*
60. *Please enter width of a rectangle*
61. *15*
62. *Area of a rectangle is 150*
63. *\*/*

**Java Class Example**

1. */\**
2. *Java Class example.*
3. *This Java class example describes how class is defined and being used*
4. *in Java language.*
6. *Syntax of defining java class is,*
7. *<modifier> class <class-name>{*
8. *// members and methods*
9. *}*
10. *\*/*
12. **public** **class** JavaClassExample{
13. */\**
14. *Syntax of defining memebers of the java class is,*
15. *<modifier> type <name>;*
16. *\*/*
17. **private** String name;
18. */\**
19. *Syntax of defining methods of the java class is,*
20. *<modifier><return-type>methodName(<optional-parameter-list>) <exception-list>{*
21. *...*
22. *}*
23. *\*/*
24. **public** **void** setName(String n){
25. *//set passed parameter as name*
26. name = n;
27. }
28. **public** String getName(){
29. *//return the set name*
30. **return** name;
31. }
32. *//main method will be called first when program is executed*
33. **public** **static** **void** main(String args[]){
34. */\**
35. *Syntax of java object creation is,*
36. *<class-name> object-name = new <class-constructor>;*
37. *\*/*
38. JavaClassExamplejavaClassExample = **new** JavaClassExample();
39. *//set name member of this object*
40. javaClassExample.setName("Visitor");
41. *// print the name*
42. System.out.println("Hello " + javaClassExample.getName());
43. }
44. }
46. */\**
47. *OUTPUT of the above given Java Class Example would be :*
48. *Hello Visitor*
49. *\*/*

**Java Factorial Example**

1. */\**
2. *Java Factorial Example*
3. *This Java Factorial Example shows how to calculate factorial of*
4. *a given number using Java.*
5. *\*/*
7. **public** **class** NumberFactorial {
9. **public** **static** **void** main(String[] args) {
11. **int** number = 5;
13. */\**
14. *\* Factorial of any number is !n.*
15. *\* For example, factorial of 4 is 4\*3\*2\*1.*
16. *\*/*
18. **int** factorial = number;
20. **for**(**int** i =(number - 1); i > 1; i--)
21. {
22. factorial = factorial \* i;
23. }
25. System.out.println("Factorial of a number is " + factorial);
26. }
27. }
29. */\**
30. *Output of the Factorial program would be*
31. *Factorial of a number is 120*
32. *\*/*

**Swap Numbers Without Using Third Variable Java Example**

1. */\**
2. *Swap Numbers Without Using Third Variable Java Example*
3. *This Swap Numbers Java Example shows how to*
4. *swap value of two numbers without using third variable using java.*
5. *\*/*
7. **public** **class** SwapElementsWithoutThirdVariableExample {
9. **public** **static** **void** main(String[] args) {
11. **int** num1 = 10;
12. **int** num2 = 20;
14. System.out.println("Before Swapping");
15. System.out.println("Value of num1 is :" + num1);
16. System.out.println("Value of num2 is :" +num2);
18. *//add both the numbers and assign it to first*
19. num1 = num1 + num2;
20. num2 = num1 - num2;
21. num1 = num1 - num2;
23. System.out.println("Before Swapping");
24. System.out.println("Value of num1 is :" + num1);
25. System.out.println("Value of num2 is :" +num2);
26. }

29. }
31. */\**
32. *Output of Swap Numbers Without Using Third Variable example would be*
33. *Before Swapping*
34. *Value of num1 is :10*
35. *Value of num2 is :20*
36. *Before Swapping*
37. *Value of num1 is :20*
38. *Value of num2 is :10*
39. *\*/*

**Java Factorial Using Recursion Example**

**Submitted By: MohitGarg**

1. */\**
2. *Java Factorial Using Recursion Example*
3. *This Java example shows how to generate factorial of a given number*
4. *using recursive function.*
5. *\*/*
7. **import** java.io.BufferedReader;
8. **import** java.io.IOException;
9. **import** java.io.InputStreamReader;
11. **public** **class** JavaFactorialUsingRecursion {
13. **public** **static** **void** main(String args[]) **throws** NumberFormatException,IOException{
15. System.out.println("Enter the number: ");
17. *//get input from the user*
18. BufferedReader br=**new** BufferedReader(**new**InputStreamReader(System.in));
19. **int** a = Integer.parseInt(br.readLine());
21. *//call the recursive function to generate factorial*
22. **int** result= fact(a);

25. System.out.println("Factorial of the number is: " + result);
26. }
28. **static** **int** fact(**int** b)
29. {
30. **if**(b <= 1)
31. *//if the number is 1 then return 1*
32. **return** 1;
33. **else**
34. *//else call the same function with the value - 1*
35. **return** b \* fact(b-1);
36. }
37. }
39. */\**
40. *Output of this Java example would be*
42. *Enter the number:*
43. *5*
44. *Factorial of the number is: 120*
45. *\*/*

**Reverse Number using Java**

1. */\**
2. *Reverse Number using Java*
3. *This Java Reverse Number Example shows how to reverse a given number.*
4. *\*/*
6. **public** **class** ReverseNumber {
8. **public** **static** **void** main(String[] args) {
10. *//original number*
11. **int** number = 1234;
12. **int** reversedNumber = 0;
13. **int** temp = 0;
15. **while**(number > 0){
17. *//use modulus operator to strip off the last digit*
18. temp = number%10;
20. *//create the reversed number*
21. reversedNumber = reversedNumber \* 10 + temp;
22. number = number/10;
24. }
26. *//output the reversed number*
27. System.out.println("Reversed Number is: " + reversedNumber);
28. }
29. }
31. */\**
32. *Output of this Number Reverse program would be*
33. *Reversed Number is: 4321*
34. *\*/*

## Java programming source code

**import**java.util.Scanner;

**class**FloydTriangle

{

**publicstaticvoid** main(Stringargs[])

{

**int** n, num=1, c, d;

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

System.out.println("Enter the number of rows of floyd's triangle you want");

n =in.nextInt();

System.out.println("Floyd's triangle :-");

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

{

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

{

System.out.print(num+" ");

num++;

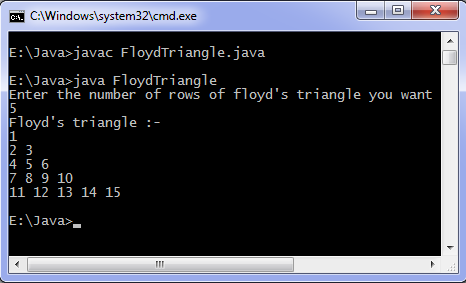
}

System.out.println();

}

}

}



Java program to convert Fahrenheit to Celsius: This code does temperature conversion from Fahrenheit scale to Celsius scale.

## Java programming code

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

**class**FahrenheitToCelsius{

**publicstaticvoid** main(String[]args){

**float**temperatue;

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

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

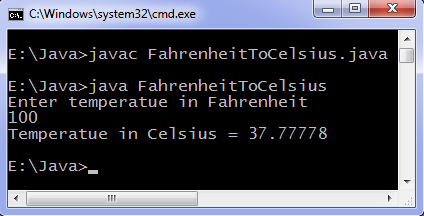
temperatue=in.nextInt();

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

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

}

}



For Celsius to Fahrenheit conversion use  
T = 9\*T/5 + 32  
where T is temperature on Celsius scale. Create and test Fahrenheit to Celsius program yourself for practice.

## Java Method example program

**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*

**publicstaticvoid** main(String[]args){

staticMethod();

Methods object =**new**Methods();

object.nonStaticMethod();

}

*// Static method*

**staticvoid**staticMethod(){

System.out.println("Static method can be called without creating object");

}

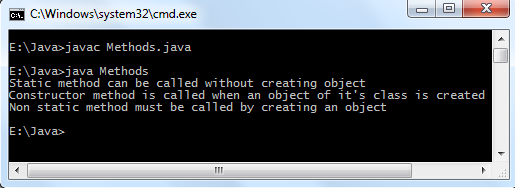
*// Non static method*

**void**nonStaticMethod(){

System.out.println("Non static method must be called by creating an object");

}

}



### Java string class program

**class**StringMethods

{

**publicstaticvoid** main(Stringargs[])

{

**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*

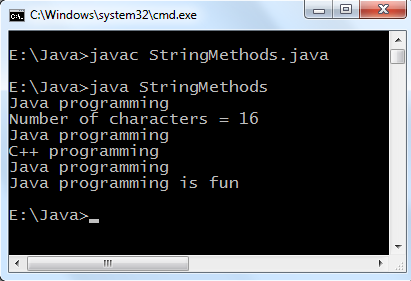
u =s.concat(" is fun");

System.out.println(s);

System.out.println(u);

}

}



## Java static block program

**class**StaticBlock{

**publicstaticvoid** main(String[]args){

System.out.println("Main method is executed.");

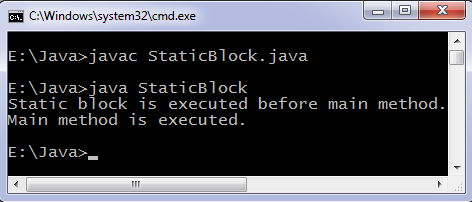
}

**static**{

System.out.println("Static block is executed before main method.");

}

}



Static block can be used to check conditions before execution of main begin, Suppose we have developed an application which runs only on Windows operating system then we need to check what operating system is installed on user machine. In our java code we check what operating system user is using if user is using operating system other than "Windows" then the program terminates.

**class**StaticBlock{

**publicstaticvoid** main(String[]args){

System.out.println("You are using Windows\_NT operating system.");

}

**static**{

Stringos=System.getenv("OS");

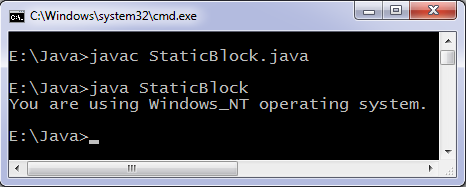
**if**(os.equals("Windows\_NT")!=**true**){

System.exit(1);

}

}

}



## Using two classes in Java program

**class** Computer {

Computer(){

System.out.println("Constructor of Computer class.");

}

**void**computer\_method(){

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

}

**publicstaticvoid** 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.");

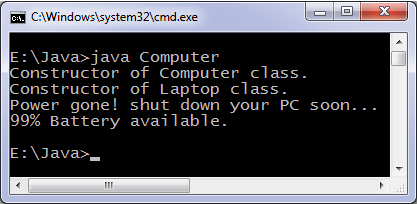
}

**void**laptop\_method(){

System.out.println("99% Battery available.");

}

}



Find Largest and Smallest Number in an Array Example

public class FindLargestSmallestNumber {

public static void main(String[] args) {

//array of 10 numbers

int numbers[] = new int[]{32,43,53,54,32,65,63,98,43,23};

//assign first element of an array to largest and smallest

int smallest = numbers[0];

intlargetst = numbers[0];

for(int i=1; i<numbers.length; i++)

{

if(numbers[i] >largetst)

largetst = numbers[i];

else if (numbers[i] < smallest)

smallest = numbers[i];

}

System.out.println("Largest Number is : " + largetst);

System.out.println("Smallest Number is : " + smallest);

}

}

/\*

Output of this program would be

Largest Number is : 98

Smallest Number is : 23

\*/

Java program to check armstrong number

This java program checks if a number is Armstrong or not. Armstrong number is a number which is equal to sum of digits raise to the power total number of digits in the number. Some Armstrong numbers are: 0, 1, 4, 5, 9, 153, 371, 407, 8208 etc.

Java programming code

importjava.util.Scanner;

classArmstrongNumber

{

public static void main(String args[])

{

int n, sum = 0, temp, remainder, digits = 0;

Scanner in = new Scanner(System.in);

System.out.println("Input a number to check if it is an Armstrong number");

n = in.nextInt();

temp = n;

// Count number of digits

while (temp != 0) {

digits++;

temp = temp/10;

}

temp = n;

while (temp != 0) {

remainder = temp%10;

sum = sum + power(remainder, digits);

temp = temp/10;

}

if (n == sum)

System.out.println(n + " is an Armstrong number.");

else

System.out.println(n + " is not an Armstrong number.");

}

staticint power(int n, int r) {

int c, p = 1;

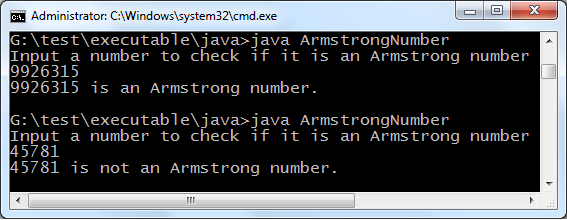
for (c = 1; c <= r; c++)

p = p\*n;

return p;

}

}



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

importjava.util.\*;

classPrimeNumbers

{

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++;

}

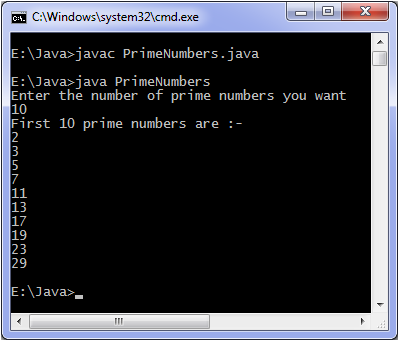
status = 1;

num++;

}

}

}



Java program to compare two strings

This program compare strings i.e test whether two strings are equal or not, compareTo method of String class is used to test equality of two String class objects. compareTo method is case sensitive i.e "java" and "Java" are two different strings if you use compareTo method. If you wish to compare strings but ignoring the case then use compareToIgnoreCasemethod.Java programming code

importjava.util.Scanner;

classCompareStrings

{

public static void main(String args[])

{

String s1, s2;

Scanner in = new Scanner(System.in);

System.out.println("Enter the first string");

s1 = in.nextLine();

System.out.println("Enter the second string");

s2 = in.nextLine();

if ( s1.compareTo(s2) > 0 )

System.out.println("First string is greater than second.");

else if ( s1.compareTo(s2) < 0 )

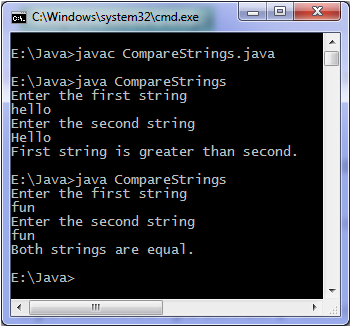
System.out.println("First string is smaller than second.");

else

System.out.println("Both strings are equal.");

}

}



String 'hello' is greater than 'Hello' as ASCII value of 'h' is greater than 'H'. To check two strings for equality you can use equals method which returns true if strings are equal otherwise false.

## Java static method example program

**class** Languages {

**publicstaticvoid** main(String[]args){

display();

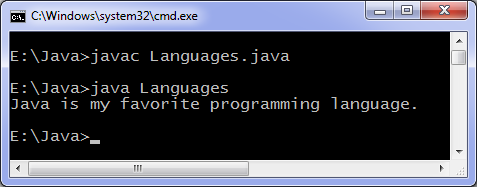
}

**staticvoid** display(){

System.out.println("Java is my favorite programming language.");

}

}



## Java static method vs instance method

Instance method requires an object of its class to be created before it can be called while static method doesn't require object creation.

**class** Difference {

**publicstaticvoid** main(String[]args){

display();*//calling without object*

Difference t =**new**Difference();

t.show();*//calling using object*

}

**staticvoid** display(){

System.out.println("Programming is amazing.");

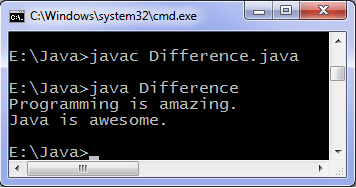
}

**void** show(){

System.out.println("Java is awesome.");

}

}



### Using static method of another classes

If you wish to call static method of another class then you have to write class name while calling static method as shown in example below.

**import**java.lang.Math;

**class** Another {

**publicstaticvoid** main(String[]args){

**int** result;

result=Math.min(10, 20);*//calling static method min by writing class name*

System.out.println(result);

System.out.println(Math.max(100, 200));

}

}

Output of program:

10

200

Here we are using min and max methods of Math class, min returns minimum of two integers and max returns maximum of two integers. Following will produce an error:

min();

We need to write class name because many classes may have a method with same name which we are calling.