

## JAVA BASIC PROGRAMMING EXAMPLES

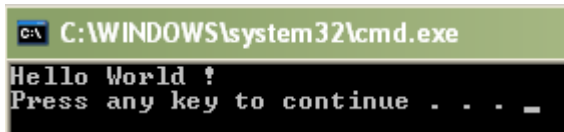
---

### Example 1: Hello World

```

1. // Java Hello World example.
2.
3. public class HelloWorldExample
4. {
5.     public static void main(String args[])
6.     {
7.
8.         //Use System.out.println() to print on console.
9.
10.        System.out.println("Hello World !");
11.
12.    }
13. }
```

### Output



```

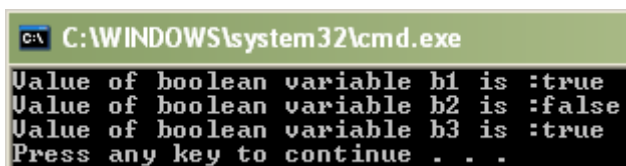
C:\ C:\WINDOWS\system32\cmd.exe
Hello World !
Press any key to continue . . . _
```

### Example 2: boolean data type

```

1. /*
2. This Java Example shows how to declare and use Java primitive boolean
   variable inside a java class.
3. */
4. public class JavaBooleanExample
5. {
6.     public static void main(String[] args)
7.     {
8.         /*
9.          * boolean data type can have only of two values; true or false.
10.        * All rational expressions retrun this type of value.
11.        * boolean <variable name> = <default value>;
12.        */
13.        boolean b1 = true;
14.        boolean b2 = false;
15.        boolean b3 = (10 > 2)? true:false;
16.
17.        System.out.println("Value of boolean variable b1 is : " + b1);
18.        System.out.println("Value of boolean variable b2 is : " + b2);
19.        System.out.println("Value of boolean variable b3 is : " + b3);
20.    }
21. }
```

### Output



```

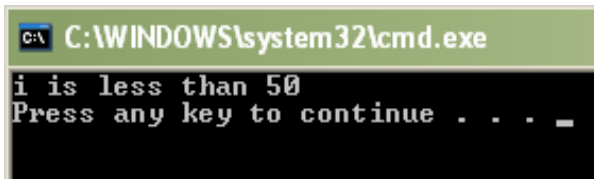
C:\ C:\WINDOWS\system32\cmd.exe
Value of boolean variable b1 is :true
Value of boolean variable b2 is :false
Value of boolean variable b3 is :true
Press any key to continue . . .
```

**Example 3: If else if else**

```

1.  /* This Java Example shows how to use if else-if statement in Java
    program.*/
2.
3.  public class IfElseIfElseExample
4.  {
5.
6.      public static void main(String[] args)
7.      {
8.
9.          /*
10.         * If Else-if statement is used to execute multiple of actions based upon
11.         * multiple conditions.
12.         * Sysntax of If Else-If statement is
13.         *
14.         * if(<condition1>)
15.         *     statement1
16.         * else if(<condition2>)
17.         *     statement2
18.         * ..
19.         * else
20.         *     statement3
21.         *
22.         * If <condition1> is true, statement1 will be executed, else if <condition2>
23.         * is true statement2 is executed and so on. If no condition is
24.         * true, then else statement will be executed.
25.         */
26.
27.         int i = 10;
28.
29.         if(i > 100)
30.             System.out.println("i is grater than 100");
31.         else if(i > 50)
32.             System.out.println("i is grater than 50");
33.         else
34.             System.out.println("i is less than 50");
35.     }
36. }

```

**Output**


```

C:\WINDOWS\system32\cmd.exe
i is less than 50
Press any key to continue . . . _

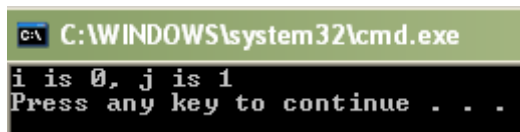
```

**Example 4: Nested Switch Statements**

```

1.  /*This example shows how to use nested switch statements in a
2.   java program.*/
3.
4.  public class NestedSwitchExample
5.  {
6.      public static void main(String[] args)
7.      {
8.
9.          /*
10.           * Like any other Java statements, switch statements
11.           * can also be nested in each other as given in below example.
12.          */
13.
14.          int i = 0;
15.          int j = 1;
16.
17.          switch(i)
18.          {
19.              case 0:
20.                  switch(j)
21.                  {
22.                      case 0:
23.                          System.out.println("i is 0, j is 0");
24.                          break;
25.
26.                      case 1:
27.                          System.out.println("i is 0, j is 1");
28.                          break;
29.
30.                      default:
31.                          System.out.println("nested default case!!");
32.                      }
33.                  break;
34.
35.                  default:
36.                      System.out.println("No matching case found!!");
37.
38.              }
39.          }
40.      }

```

**Output**


The screenshot shows a Windows command prompt window with the title bar "C:\WINDOWS\system32\cmd.exe". The command prompt displays the output of the Java program: "i is 0, j is 1" followed by "Press any key to continue . . .".

**Example 5: Arithmetic Operators**

```

1.  /* This example shows how to use Java arithmetic operators like +
    (addition), - (subtraction), * (multiplication) and / (division).
2.  */
3.
4.  public class ArithmeticOperatorsExample
5.  {
6.
7.      public static void main(String[] args)
8.      {
9.
10.         System.out.println("Arithmetic operators example :");
11.
12.         int i = 50 + 20;
13.         int j = i - 10;
14.         int k = j * 2;
15.         double l = k / 6;
16.
17.         System.out.println("i = " + i);
18.         System.out.println("j = " + j);
19.         System.out.println("k = " + k);
20.         System.out.println("l = " + l);
21.     }
22. }

```

**Output**

```

C:\WINDOWS\system32\cmd.exe
Arithmetic operators example :
i = 70
j = 60
k = 120
l = 20.0
Press any key to continue . . . _

```

**Example 6: Declare multiple variables in for loop**

```

1.  /*This Java Example shows how to declare multiple variables in Java For
    loop using declaration block.*/
2.
3.  public class DeclaringMultipleVariablesInForLoopExample
4.  {
5.
6.      public static void main(String[] args)
7.      {
8.          /* Multiple variables can be declared in declaration block of for loop.*/
9.          for(int i=0, j=1, k=2 ; i<5 ; i++)
10.             System.out.println("I : " + i + ",j : " + j + ", k : " + k);
11.     }
12. }

```

**Output**

```

C:\WINDOWS\system32\cmd.exe
I : 0,j : 1, k : 2
I : 1,j : 1, k : 2
I : 2,j : 1, k : 2
I : 3,j : 1, k : 2
I : 4,j : 1, k : 2
Press any key to continue . . . _

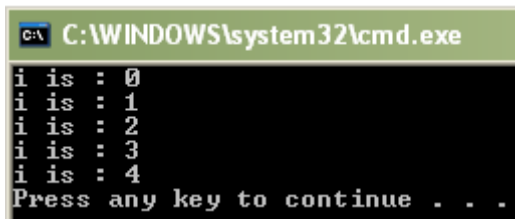
```

**Example 7: Do While Loop**

```

1.  /* This Java Example shows how to use do while loop to iterate in Java
    program.*/
2.
3.  public class DoWhileExample
4.  {
5.      public static void main(String[] args)
6.      {
7.          // Do while loop executes statment until certain condition become false.
8.
9.          int i =0;
10.
11.          do
12.          {
13.              System.out.println("i is : " + i);
14.              i++;
15.
16.          }while(i < 5);
17.      }
18.  }

```

**Output**


```

C:\WINDOWS\system32\cmd.exe
i is : 0
i is : 1
i is : 2
i is : 3
i is : 4
Press any key to continue . . .

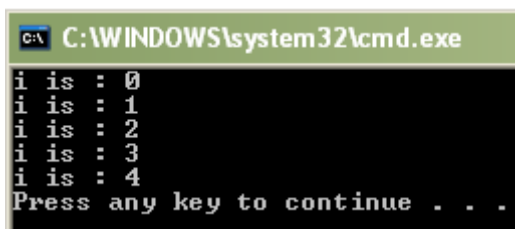
```

**Example 8: While Loop**

```

1.  //This Java Example shows how to use while loop to iterate in Java program.
2.  public class SimpleWhileLoopExample
3.  {
4.      public static void main(String[] args)
5.      {
6.          int i = 0;
7.
8.          while(i < 5)
9.          {
10.              System.out.println("i is : " + i);
11.              i++;
12.          }
13.      }
14.  }

```

**Output**


```

C:\WINDOWS\system32\cmd.exe
i is : 0
i is : 1
i is : 2
i is : 3
i is : 4
Press any key to continue . . .

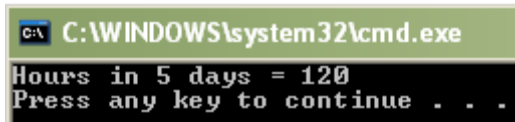
```

**Example 9: Using final variable**

```

1. /*This Java Example shows how to declare and use final variable in
2.    a java class.*/
3.
4. public class FinalVariableExample
5. {
6.
7.     public static void main(String[] args)
8.     {
9.
10.        /*
11.         * Final variables can be declared using final keyword.
12.         * Once created and initialized, its value can not be changed.
13.         */
14.
15.        final int hoursInDay=24;
16.
17.        //This statement will not compile. Value can't be changed.
18.        //hoursInDay=12;
19.
20.        System.out.println("Hours in 5 days = " + hoursInDay * 5);
21.    }
22. }

```

**Output**


```

C:\WINDOWS\system32\cmd.exe
Hours in 5 days = 120
Press any key to continue . . .

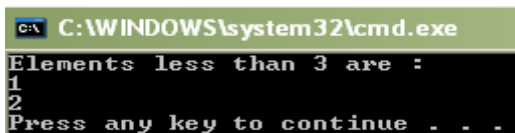
```

**Example 10: Using break statement**

```

1. //This example shows how to use java break statement.
2. public class JavaBreakExample
3. {
4.     public static void main(String[] args)
5.     {
6.
7.         int intArray[] = new int[]{1,2,3,4,5};
8.
9.         System.out.println("Elements less than 3 are : ");
10.        for(int i=0; i < intArray.length ; i ++)
11.        {
12.            if(intArray[i] == 3)
13.                break;
14.            else
15.                System.out.println(intArray[i]);
16.        }
17.    }
18. }

```

**Output**


```

C:\WINDOWS\system32\cmd.exe
Elements less than 3 are :
1
2
Press any key to continue . . .

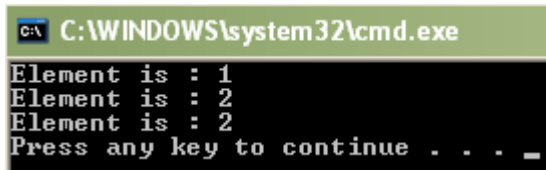
```

**Example 11: Using continue statement with label**

```

1.  /*
2.  This example shows how to use java continue statement to skip to next
    iteration of the labeled loop.*/
3.
4.  public class JavaContinueWithLabelExample
5.  {
6.      public static void main(String[] args)
7.      {
8.
9.          int intArray[][] = new int[][]{{1,2},{2,3}};
10.
11.         Outer:
12.         for(int i=0; i < intArray.length; i++)
13.         {
14.             for(int j=0; j < intArray[i].length ; j++)
15.             {
16.                 if(intArray[i][j] == 3)
17.                     continue Outer;
18.                 System.out.println("Element is : " + intArray[i][j]);
19.             }
20.         }
21.     }
22. }

```

**Output**


```

C:\WINDOWS\system32\cmd.exe
Element is : 1
Element is : 2
Element is : 2
Press any key to continue . . . _

```

**Example 12: Using String class, Simple String**

```

1.  // Describes how Java String object is created and used.
2.  public class JavaStringExample
3.  {
4.      public static void main(String args[])
5.      {
6.          // creates new empty string.
7.          String str1 = new String("");
8.
9.          //creates new string object whose content would be Hello World
10.         String str2 = new String("Hello world");
11.
12.         //creates new string object whose content would be Hello World
13.         String str3 = "Hello Wolrd";
14.
15.         System.out.println( str1.length());
16.     }
17. }

```

**Output**


```

0
Press any key to continue . . . _

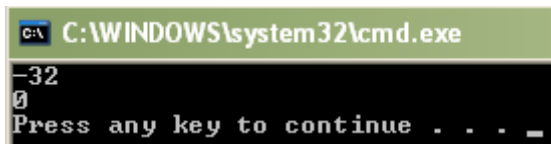
```

**Example 13: Using String class, Comparing two Strings**

```

1.  /*
2.  Describes how Java String is compared with another Java String object.
3.  */
4.
5.  public class JavaStringCompareExample
6.  {
7.      public static void main(String args[])
8.      {
9.
10.         String str = "Hello World";
11.         String anotherString = "hello world";
12.         Object objStr = str;
13.
14.         /* compare two strings, case sensitive */
15.         System.out.println( str.compareTo(anotherString) );
16.
17.         /* compare two strings, ignores character case */
18.         System.out.println( str.compareToIgnoreCase(anotherString) );
19.
20.     }
21. }

```

**Output**


```

C:\WINDOWS\system32\cmd.exe
-32
0
Press any key to continue . . . _

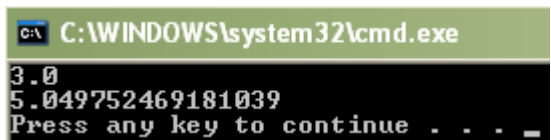
```

**Example 14: Using Math class, Square root of a number using Math.sqrt**

```

1.  /*
2.  How to find square root of a number using sqrt method of Math class.
3.  */
4.  public class FindSquareRootExample
5.  {
6.      public static void main(String[] args)
7.      {
8.          //returns square root of 9, i.e. 3
9.          System.out.println(Math.sqrt(9));
10.
11.         //returns square root of 25.5
12.         System.out.println(Math.sqrt(25.5));
13.     }
14. }

```

**Output**


```

C:\WINDOWS\system32\cmd.exe
3.0
5.049752469181039
Press any key to continue . . . _

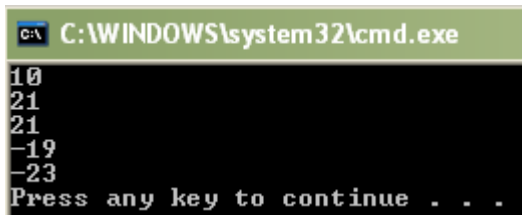
```

**Example 15: Using Math class, Round numbers using Math.round**

```

1.  /*
2.  How to round given float or double number using round method of Java
    Math class.
3.  */
4.  public class RounFloatDoubleNumbersExample
5.  {
6.
7.      public static void main(String[] args)
8.      {
9.
10.         /*
11.          * To round float number, use
12.          * static int round(float f) method of Java Math class.
13.          *
14.          * It returns closest int number to the argument.
15.          * Internally, it adds 0.5 to the argument, takes floor value and
16.          * casts the result into int.
17.          * i.e. result = (int) Math.floor( argument value + 0.5f )
18.          */
19.
20.          //returns same value
21.          System.out.println(Math.round(10f));
22.
23.          // returns (int) Math.floor(10.6) = 10
24.          System.out.println(Math.round(20.5f));
25.
26.          //returns (int) Math.floor(20.5 + 0.5) = 30
27.          System.out.println(Math.round(20.5f));
28.
29.          //returns (int) Math.floor(-18.9) = 19
30.          System.out.println(Math.round(-19.4f));
31.
32.          //returns (int) Math.floor(-23) = -23
33.          System.out.println(Math.round(-23.5f));
34.      }
35.  }

```

**Output**


```

C:\WINDOWS\system32\cmd.exe
10
21
21
-19
-23
Press any key to continue . . .

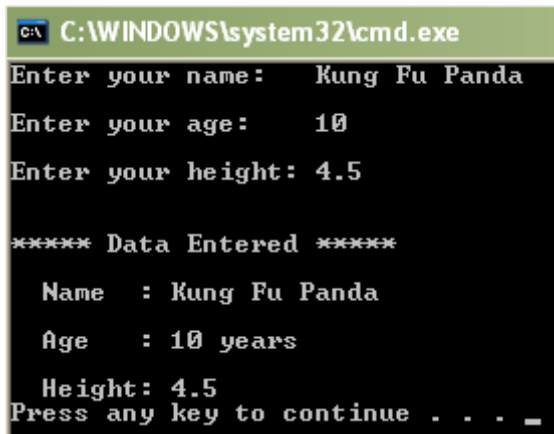
```

**Example 16: Using Scanner class to read input from console**

```

1.  /*
2.  How to use Scanner class to take input from console
3.  */
4.  import java.util.*;
5.  public class UsingScanner
6.  {
7.      public static void main(String[] args)
8.      {
9.          //Creating Scanner Object
10.         Scanner sc = new Scanner(System.in);
11.
12.         //reading String Line
13.         System.out.print("Enter your name:   ");
14.         String strname = sc.nextLine();
15.         System.out.println();
16.         //reading int
17.         System.out.print("Enter your age:     ");
18.         int age = sc.nextInt();
19.         System.out.println();
20.         //reading float
21.         System.out.print("Enter your height: ");
22.         float height = sc.nextFloat();
23.         System.out.print("\n\n***** Data Entered *****\n\n");
24.         System.out.println("  Name   : " + strname + "\n");
25.         System.out.println("  Age    : " + age + " years\n");
26.         System.out.println("  Height: " + height);
27.     }
28. }

```

**Output**


```

C:\WINDOWS\system32\cmd.exe
Enter your name:   Kung Fu Panda
Enter your age:    10
Enter your height: 4.5

***** Data Entered *****

  Name   : Kung Fu Panda
  Age    : 10 years
  Height: 4.5
Press any key to continue . . . _

```

😊😊😊😊 THE END 😊😊😊😊