

Java Guide



Meher Krishna Patel

Created on : October, 2017

Last updated : October, 2018

Table of contents

Table of contents	i
1 Java	1
1.1 Hello World	1
1.2 Variables	1
1.3 Printing values in Binary, Hex, Octal and Decimal formats	2
1.4 Define constant using 'final'	2
1.5 Read input from user	3
1.6 Operators and Control statements	4
1.6.1 Operators	4
1.6.2 Decision statements	4
1.6.3 Loops	6
1.6.4 Continue and Break statements	7
1.7 Array	8
1.8 OOPs	9
1.8.1 Class and object	9
1.8.2 Method with parameters	10
1.8.3 set and get method	10
1.8.4 Constructor	11
1.8.5 Inheritance	12
1.8.6 multiple constructor	13
1.8.7 Polymorphism	14
1.8.8 Abstract class	15

Chapter 1

Java

1.1 Hello World

Warning: Filename should be same as class-name, otherwise compilation error will be generated.

```
// HelloWorld.java

public class HelloWorld{

    // main method is the starting point for execution
    public static void main(String[] args){
        System.out.println("Hello World"); // println = print + \n
    }
}
```

```
$ javac HelloWorld.java
$ java HelloWorld
Hello World
```

1.2 Variables

```
// HelloWorld.java

public class HelloWorld{
    public static void main(String[] args){
        System.out.println("Hello World");

        int i = 100;
        double j = 10.10;
        double k;

        k = i + j;

        // System.out.println("k= " + k);
        System.out.print("k= " + k + "\n");

        char c = 'c'; // char in single quote
        System.out.println("char = " + c);
    }
}
```

(continues on next page)

(continued from previous page)

```

boolean b = true; // true/false
System.out.println("Boolean b = " + b);
System.out.println("3>4 : " + (3>4));

// string is not a datatype; it's a predefined class
String name = "Meher Krishna Patel"; // string in double quote
System.out.println("Name = " + name);
}
}

```

```

$ javac HelloWorld.java
$ java HelloWorld
Hello World
k= 110.1
char = c
Boolean b = true
3>4 :false
Name = Meher Krishna Patel

```

1.3 Printing values in Binary, Hex, Octal and Decimal formats

```

// hexDec.java
class hexDec{
    public static void main(String arg[]){
        int x = 15;
        System.out.printf("x = %d\n", x);

        int h = 0xb;
        int o = 013;

        System.out.printf(" hex value of x = %x\n", x); // hex
        System.out.printf(" oct value of x = %o\n", x); // octal
        System.out.printf(" decimal value of x = %d\n\n", x); // decimal

        System.out.printf(" hex value of h = %x\n", h); // hex
        System.out.printf(" oct value of h = %o\n", h); // octal
        System.out.printf(" decimal value of h = %d\n\n", h); // decimal

        System.out.printf(" hex value of o = %x\n", o); // hex
        System.out.printf(" oct value of o = %o\n", o); // octal
        System.out.printf(" decimal value of o = %d\n\n", o); // decimal

    }
}

```

1.4 Define constant using 'final'

Warning: Since method 'main' is of type **static**, therefore variables 'pi' and 'blob' should be of type **'static'**, otherwise these can not be used in 'main' function.

```
// defineConst.java

public class defineConst{
    public static final double pi = 3.14;
    public static final String blog = "PythonDSP";

    public static void main(String[] args){
        final int radius = 2;
        double area;

        area = pi * Math.pow(radius, 2);
        System.out.printf("Area of circle = %f\n", area);
        System.out.printf("Blog : %s\n", blog);
    }
}
```

```
$ javac defineConst.java
$ java defineConst
Area of circle = 12.560000
Blog : PythonDSP
```

```
$ javac hexDec.java
$ java hexDec
x = 15
hex value of x = f
oct value of x = 17
decimal value of x = 15

hex value of h = b
oct value of h = 13
decimal value of h = 11

hex value of o = b
oct value of o = 13
decimal value of o = 11
```

1.5 Read input from user

```
// ReadEx.java

import java.util.Scanner; // used to read input

public class ReadEx{
    public static void main(String[] args){
        // Create object of Scanner
        Scanner scn = new Scanner(System.in);

        int x, y, z; // variables

        // print
        System.out.print("Enter first number: ");
        x = scn.nextInt(); // read x

        System.out.print("Enter second number: ");
        y = scn.nextInt(); // read y

        z = x+y; // add
    }
}
```

(continues on next page)

(continued from previous page)

```

    // printf
    System.out.printf("Sum = %d\n", z);
}
}

```

1.6 Operators and Control statements

1.6.1 Operators

1.6.1.1 Arithmetic operators

```

// incrementEx.java

public class incrementEx{
    public static void main(String[] args){
        int i = 1;
        System.out.printf("i = %d\n\n", i);

        // print first, then increment
        System.out.printf("i++ = %d (i.e. access first and then increment): \n", i++); // 1
        System.out.printf("i = %d\n\n", i); // 2 (i.e. i is incremented by above statement)

        // increment first, then print
        System.out.printf("++i = %d (i.e. increment first and then access):\n", ++i); // 3
        System.out.printf("i = %d\n", i); // 3
    }
}

```

```

$ java incrementEx
i = 1

i++ = 1 (i.e. access first and then increment):
i = 2

++i = 3 (i.e. increment first and then access):
i = 3

```

1.6.1.2 Relational operators

1.6.1.3 Logical operators

1.6.2 Decision statements

1.6.2.1 If-else statements

```

// ElseIfEx.java

import java.util.Scanner; // used to read input

public class ElseIfEx{
    public static void main(String[] args){
        // Create object of Scanner
        Scanner scn = new Scanner(System.in);
    }
}

```

(continues on next page)

(continued from previous page)

```

System.out.print("Enter Grade: ");
String input = scn.next();
char grade = input.charAt(0);

// if grade is A or a
if (grade == 'A' || grade == 'a') // logical operator
    System.out.printf("Excellent Student\n");
else if ((grade == 'B') || (grade == 'b'))
    System.out.printf("Very Good Student\n");
else if (grade == 'C' || grade == 'c')
    System.out.printf("Good Student\n");
else if (grade == 'D' || grade == 'd')
    System.out.printf("Need improvements\n");
else
    System.out.printf("Invalid grade\n");
}
}

```

```

$ javac ElseIfEx.java
$ java ElseIfEx
Enter Grade: b
Very Good Student
$ java ElseIfEx
Enter Grade: 4
Invalid grade

```

1.6.2.2 Switch case

```

// switchCase.java

import java.util.Scanner; // used to read input

public class switchCase{
    public static void main(String[] args){
        // Create object of Scanner
        Scanner scn = new Scanner(System.in);

        System.out.print("Enter Grade: ");
        String input = scn.next();
        char grade = input.charAt(0);

        switch(grade){
            case ('A'):
                System.out.printf("A: performance is excellent\n");
                break;
            case 'B' :
                System.out.printf("B: performance is good\n");
                break;
            case 'C' :
                System.out.printf("C: performance is not bad\n");
                break;
            default :
                System.out.printf("Invalid grade\n");
        }
    }
}

```

```
$ javac switchCase.java
```

(continues on next page)

(continued from previous page)

```
$ java switchCase
Enter Grade: A
A: performance is excellent
$ java switchCase
Enter Grade: a
Invalid grade
```

1.6.2.3 Conditional operator (?:)

1.6.3 Loops

1.6.3.1 For loop

```
// forLoop.java

public class forLoop{
    public static void main(String[] args){
        int i;

        // print 0 to 4
        for (i=0; i<5; i++){
            System.out.printf("%d, ", i); //0, 1, 2, 3, 4,
        }

        // value of i is 5
        System.out.printf("\n");
        System.out.printf("Value of i = %d\n", i); // Value of i = 5
    }
}
```

```
$ javac forLoop.java
$ java forLoop
0, 1, 2, 3, 4,
Value of i = 5
```

1.6.3.2 While loop

```
// whileLoop.java

public class whileLoop{
    public static void main(String[] args){
        int i=0;

        // print 0 to 4
        while(i < 5){
            System.out.printf("%d, ", i); //0, 1, 2, 3, 4,
            i = i+1;
        }

        // value of i is 5
        System.out.printf("\n");
        System.out.printf("Value of i = %d\n", i); // Value of i = 5
    }
}
```



```
$ javac whileLoop.java
$ java whileLoop
0, 1, 2, 3, 4,
Value of i = 5
```

1.6.3.3 Do while loop

```
// dowhileLoop.java

public class dowhileLoop{
    public static void main(String[] args){
        int i=0;

        // print 0 to 4
        do{
            System.out.printf("%d, ", i); //0, 1, 2, 3, 4,
            i = i+1;
        }while(i < 5);

        // value of i is 5
        System.out.printf("\n");
        System.out.printf("Value of i = %d\n", i); // Value of i = 5
    }
}
```

```
$ javac dowhileLoop.java
$ java dowhileLoop
0, 1, 2, 3, 4,
Value of i = 5
```

1.6.4 Continue and Break statements

1.6.4.1 Continue

```
// forContinueLoop.java

public class forContinueLoop{
    public static void main(String[] args){
        int i;

        // print 0 to 4
        for (i=0; i<5; i++){
            if (i%2==0)
                continue;
            System.out.printf("%d, ", i); //0, 1, 2, 3, 4,

        }

        // value of i is 5
        System.out.printf("\n");
        System.out.printf("Value of i = %d\n", i); // Value of i = 5
    }
}
```

```
$ javac forContinueLoop.java
$ java forContinueLoop
```

(continues on next page)

(continued from previous page)

```
1, 3,
Value of i = 5
```

1.6.4.2 Break

```
// whileBreakLoop.java

public class whileBreakLoop{
    public static void main(String[] args){
        int i=0;

        // print 0 to 6 (break at 7)
        while(i < 15){
            System.out.printf("%d, ", i);
            i = i+1;
            if (i%7==0)
                break;
        }

        System.out.printf("\n");
        System.out.printf("Value of i = %d\n", i);
    }
}
```

```
$ javac whileBreakLoop.java
$ java whileBreakLoop
0, 1, 2, 3, 4, 5, 6,
Value of i = 7
```

1.7 Array

```
// arrayEx.java

public class arrayEx{
    public static void main(String[] args){
        int i;

        int[] a; // uninitialized array of size 5 i.e. a[0]-a[4]
        a = new int[10];

        double[] b={2, 4.5, 6}; // initialized array of size 3
        int[] c = {3, 5};

        System.out.printf("a[1] = %d\n", a[1]); // uninitialized array has 0 value

        System.out.printf("c[0] = %d\n", c[0]); // 3

        // print all values of array b
        // %10s create the width of 10 after 'element'
        // and 'value' will be printed as right-aligned e.g. see 4.5 in output
        System.out.printf("%s %10s\n", "element", "value");
        for (i=0; i<3; i++){
            System.out.printf("%4d %12.1f\n", i, b[i]); // %12.1f = show minimum 12 integer & 1 decimal
        }
    }
}
```

(continues on next page)

(continued from previous page)

```

// assign values of array a
for (i=0; i<5; i++){
    a[i] = 2*i;
}

// print values of array a
System.out.printf("%s %10s\n", "element", "value");
for (i=0; i<5; i++){
    System.out.printf("%4d %12d\n", i, a[i]); // %12.1f = show minimum 12 integer and 1 decimal
→place
}
}
}

```

```

$ javac arrayEx.java
$ java arrayEx
a[1] = 0
c[0] = 3
element      value
  0           2.0
  1           4.5
  2           6.0
element      value
  0           0
  1           2
  2           4
  3           6
  4           8

```

1.8 OOPs

1.8.1 Class and object

- Class Jungle

```

// Jungle.java

public class Jungle{

    // public : to allow access outside the class
    public void welcomeMessage(){
        System.out.println("Welcome to Jungle");
    }
}

```

- Object 'j' of class Jungle

```

// JungleTest.java

public class JungleTest{
    public static void main(String[] args){
        Jungle j = new Jungle();
        j.welcomeMessage();
    }
}

```

- Code execution

```
$ javac Jungle.java
$ javac JungleTest.java
$ java JungleTest
Welcome to Jungle
```

Note: We can compile multiple file as below,

```
$ javac Jungle.java JungleTest.java

or

$ javac *.java
```

1.8.2 Method with parameters

```
// Jungle.java

public class Jungle{

    // public : to allow access outside the class
    public void welcomeMessage(String name){
        System.out.printf("Hello %s! Welcome to Jungle\n", name);
    }
}
```

```
// JungleTest.java

public class JungleTest{
    public static void main(String[] args){
        Jungle j = new Jungle();
        j.welcomeMessage("Meher");
    }
}
```

```
$ javac *.java
$ java JungleTest
Hello Meher! Welcome to Jungle
```

1.8.3 set and get method

```
// Jungle.java

public class Jungle{

    private String visitorName;

    // set (save) visitor name
    public void setVisitorName(String name){
        visitorName = name;
    }

    // get visitor name
    public String getVisitorName(){
        return visitorName;
    }
}
```

(continues on next page)

(continued from previous page)

```

// print message
public void welcomeMessage(){
    System.out.printf("Hello %s! Welcome to Jungle\n", getVisitorName());
}
}

```

```

// JungleTest.java

public class JungleTest{
    public static void main(String[] args){
        Jungle j = new Jungle();

        // save visitor name
        j.setVisitorName("Krishna");

        // print message
        j.welcomeMessage();
    }
}

```

```

$ javac *.java
$ java JungleTest
Hello Krishna! Welcome to Jungle

```

1.8.4 Constructor

- Initialize object using constructor

```

// Jungle.java

public class Jungle{

    private String visitorName;

    // Constructor : name should be same as class
    public Jungle(String name){ // constructor with one argument
        visitorName = name;
    }

    // set (save) visitor name
    public void setVisitorName(String name){
        visitorName = name;
    }

    // get visitor name
    public String getVisitorName(){
        return visitorName;
    }

    // print message
    public void welcomeMessage(){
        System.out.printf("Hello %s! Welcome to Jungle\n", getVisitorName());
    }
}

```

```

// JungleTest.java

```

(continues on next page)

(continued from previous page)

```

public class JungleTest{
    public static void main(String[] args){
        Jungle j = new Jungle("Patel");

        // print message
        j.welcomeMessage();
    }
}

```

```

$ javac *.java
$ java JungleTest
Hello Patel! Welcome to Jungle

```

1.8.5 Inheritance

```

// RateJungle.java

public class RateJungle extends Jungle{
    private int feedback;
    private String name;

    public RateJungle(String name){
        super(name); // call the base class constructor
        feedback = 0; // set feedback to 0
    }

    public void setFeedback(int val){
        feedback = val;
    }

    public void printRating(){
        System.out.printf("Thanks %s\n", getVisitorName());
        System.out.printf("Your feedback is set as : %d\n", feedback);
    }
}

```

```

// RateJungleTest.java

public class RateJungleTest{
    public static void main(String[] args){
        RateJungle m = new RateJungle("Meher");
        RateJungle k = new RateJungle("Krishna");

        m.printRating();
        m.setFeedback(2);
        m.printRating();

        k.printRating();
    }
}

```

```

$ javac *.java
$ java RateJungleTest
Thanks Meher
Your feedback is set as : 0
Thanks Meher
Your feedback is set as : 2

```

(continues on next page)

(continued from previous page)

Thanks Krishna
Your feedback is set as : 0

1.8.6 multiple constructor

```
// RateJungle.java

public class RateJungle extends Jungle{
    private int feedback;
    private String name;

    // constructor 1
    public RateJungle(String name){
        super(name); // call the base class constructor
        feedback = 0; // set feedback to 0
    }

    // constructor 2
    public RateJungle(String name, int val){
        super(name);
        feedback = val;
    }

    // constructor 3 : empty constructor
    public RateJungle(){
    }

    public void setFeedback(int val){
        feedback = val;
    }

    public void printRating(){
        System.out.printf("Thanks %s\n", getVisitorName());
        System.out.printf("Your feedback is set as : %d\n", feedback);
    }
}
```

- Since child class has empty constructor, therefore we need to add empty constructor in parent class as well.

```
// Jungle.java

public class Jungle{

    private String visitorName;

    // Constructor : name should be same as class
    public Jungle(String name){ // constructor with one argument
        visitorName = name;
    }

    // empty constructor : as child class has an empty constructor
    public Jungle(){
    }

    // set (save) visitor name
    public void setVisitorName(String name){
        visitorName = name;
    }
}
```

(continues on next page)

(continued from previous page)

```

// get visitor name
public String getVisitorName(){
    return visitorName;
}

// print message
public void welcomeMessage(){
    System.out.printf("Hello %s! Welcome to Jungle\n", getVisitorName());
}
}

```

```

// RateJungleTest.java

public class RateJungleTest{
    public static void main(String[] args){
        RateJungle m = new RateJungle("Meher");
        RateJungle k = new RateJungle("Krishna", 5);
        RateJungle p = new RateJungle();

        p.setVisitorName("Patel");
        p.setFeedback(2);

        m.printRating();
        k.printRating();
        p.printRating();
    }
}

```

```

$ javac *.java
$ java RateJungleTest
Thanks Meher
Your feedback is set as : 0
Thanks Krishna
Your feedback is set as : 5
Thanks Patel
Your feedback is set as : 2

```

1.8.7 Polymorphism

```

// Animal.java

public class Animal{
    public void scarySound(){
        System.out.println("Animals are running away due to scary sound.");
    }
}

```

```

// Bird.java

public class Bird{
    public void scarySound(){
        System.out.println("Birds are flying away due to scary sound.");
    }
}

```

```

// Insect.java

```

(continues on next page)

(continued from previous page)

```
// empty class
public class Insect{
}
```

```
// PolymorphismTest.java

public class PolymorphismTest{
    public static void main(String[] args){
        Animal a = new Animal();
        Bird b = new Bird();

        a.scarySound();
        b.scarySound();
    }
}
```

```
$ javac *.java
$ java PolymorphismTest
Animals are running away due to scary sound.
Birds are flying away due to scary sound.
```

1.8.8 Abstract class

```
// Abstract_Jungle.java

// abstract class
public abstract class Abstract_Jungle{

    private String visitorName;

    // set (save) visitor name
    public void setVisitorName(String name){
        visitorName = name;
    }

    // get visitor name
    public String getVisitorName(){
        return visitorName;
    }

    // print message
    public void welcomeMessage(){
        System.out.printf("Hello %s! Welcome to Jungle\n", getVisitorName());
    }

    // abstract method
    public abstract void scarySound();
}
```

```
// Animal.java

public class Animal extends Abstract_Jungle{
    public void scarySound(){
        System.out.println("Animals are running away due to scary sound.");
    }
}
```

```
// Bird.java

public class Bird extends Abstract_Jungle{
    public void scarySound(){
        System.out.println("Birds are flying away due to scary sound.");
    }
}
```

```
// Abstract_JungleTest.java

public class Abstract_JungleTest{
    public static void main(String[] args){
        Animal a = new Animal();
        Bird b = new Bird();
        Insect i = new Insect();

        a.scarySound();
        b.scarySound();
        i.scarySound();
    }
}
```

```
$ javac *.java
$ java Abstract_JungleTest
Animals are running away due to scary sound.
Birds are flying away due to scary sound.
Insects do not care about scary sound.
```

Error: If we do not define the 'scarySound()' in class Insect, then following error will be generated.

```
$ javac *.java

Insect.java:4: error: Insect is not abstract and does not override abstract method scarySound() in
↳Abstract_Jungle
public class Insect extends Abstract_Jungle{
    ^
1 error
```