**Date:18/07/2023**

**FLOAT AND DOUBLE IN DECIMAL POINTS**

Assignment:03

Nmae: Suvarna Ballid

Email id:Suvarnaballid726@gmail.com

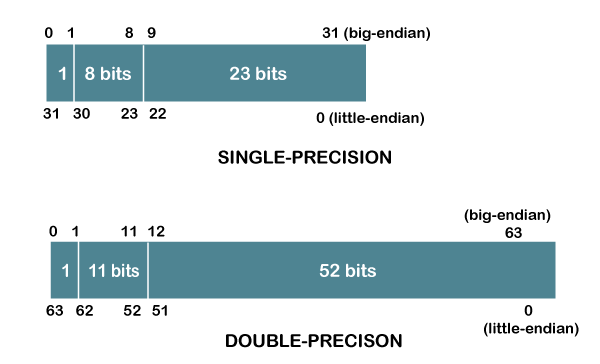
Float and double

In Java, Data types specify the size and type of values. It is used to store the floating values of an identifier. Data types are classified into two categories, primitive and non-primitive. Primitive data type includes all the predefined data types such as Integer, Character, Boolean, Floating-Point, etc. while the non-primitive data-type includes user-defined data types such as Classes, Arrays, and Interfaces, etc. Both float and double data types store floating values but still, they are different. In this section, we will learn the differences between float and double datatype in [Java](https://www.javatpoint.com/java-tutorial).

There are two types of Floating-Point data types:

* float Data Type
* double Data Type
* Single-Precision: It consists of one sign bit (S), eight exponent bits (E), and twenty-three mantissa bits (M).

Double-Precision: It consists of one sign bit (S), eleven exponent bits (E), and fifty-two mantissa bits (M). **Double-Precision:**It consists of **one** sign bit (S), **eleven** exponent bits (E), and **fifty-two** mantissa bits (M).

* float Data Type
* It is a 32-bit, single-precision [IEEE](https://www.javatpoint.com/ieee-full-form) 754 (Standard for Floating-Point Arithmetic) floating-point number. It means that it gives 6-7 decimal digits precision. It is used if we want to use memory effectively because it takes less memory in comparison to double data type. To define a float value, we must use a suffix f or F. Its default value is 0.0f. By default, float numbers are treated as double in Java.

1. Public class FloatDataTypeExample
2. {

Public static void main(String[] args)

1. {

floatx = 1.0f, y = 3.0f;

1. float=x/y;
2. System.out.println(.x/y + z);
3. }
4. }

Output; x/y = 0.33333334

double Data Type

1. The double data type is a 64-bit double-precision IEEE 754 floating-point number. It means that it gives 15-16 decimal digits precision. It consumes more memory in comparison to the float data type. It is used to store decimal values. Its default value is 0.0d. It is optional to add suffix d or D. For example: DoubleDataTypeExample
2. {
3. Public static void main(String[] args)
4. {

 x = 1.0, y = 3.0;

double z=x/y;

1. System.out.println(“x/y”=+z);
2. }
3. }

Output:x/y = 0.3333333333333333

Through the above two examples, it is clear that the double data type takes more memory to store a double-precision number and also gives the more accurate result up to 16 decimal digits. While the float data type takes less space to store single-precision numbers and it gives results up to 6 decimal places.

***Digits permitted after Decimal point:***

**The first digit after the decimal represents the tenths place. The next digit after the decimal represents the hundredths place. The remaining digits continue to fill in the place values until there are no digits left.**

 there is no limit to the number of decimal places you can write out for a number. However, in practice, the number of decimal places that can be accurately represented is limited by the precision of the mathematical representation and the storage capacity of the device used to store and process the number.

## The decimal system consists of 10 single-digit numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. The number 9 is followed by 10, which is followed by 11, then 12, and so on. The number on the left is incremented by 1 each time the digit to the right goes beyond 9.

## Place value chart

The number 87.65487.65487, point, 654 can be placed on a place value chart as follows:

| Tens | Ones | **.** | Tenths | Hundredths | Thousandths |
| --- | --- | --- | --- | --- | --- |
| 888 | 777 | ..point | 666 | 555 | 444 |

Example:

0.81=0.8+0.01

public class Decimal {

public static void main(String[] args) {

double num = 1.34567;

System.out.format("%.4f", num);

}

}

Output:1.3457

In the above program, we've used the format() method to print the given floating-point number num to 4 decimal places.  The 4 decimal places are given by the format .4f.

This means, print only up to 4 places after **the dot** (decimal places), and f means to print the floating-point number.

mport java.math.RoundingMode;

import java.text.DecimalFormat;

public class Decimal {

public static void main(String[] args) {

DecimalFormat df = new DecimalFormat("#.###");

df.setRoundingMode(RoundingMode.CEILING);

System.out.println(df.format(num));

}

}

Output:1.346

In the above program, we've used Decimal formate class to round a given number num.

We declare the format using the # patterns. This means we want num up to 3 decimal places. We also set the rounding mode to Ceiling, this causes the last given place to be rounded to its next number.

So, 1.34567 rounded to 3 decimal places prints 1.346, 6 is the next number for 3rd place decimal 5.