



Courses Upto 25% Off DSA Data Structures Algorithms Interview Preparation Data Science Topi

Difficulty Level : Basic • Last Updated : 03 Apr, 2023

Read

Discuss

Courses

Practice

Video

Double class is a wrapper class for the primitive type double which contains several methods to effectively deal with a double value like converting it to a string representation, and vice-versa. An object of the Double class can hold a single double value. **Double class** is a wrapper class for the primitive type double which contains several methods to effectively deal with a double value like converting it to a string representation, and vice-versa. An object of the Double class can hold a single double value.

There are mainly two constructors to initialize a Double-object.

A. Double(double b): Creates a Double-object initialized with the value provided where it takes a value with which to initialize as a parameter.

```
public Double(double d)
```

Parameters: Value with which to initialize

B. Double(String s): Creates a Double-object initialized with the parsed double value provided by string representation where it takes a string representation of the byte value as a parameter.

Default radix is taken to be 10.

```
public Double(String s) throws NumberFormatException
```

Methods of Double Class

Method	Action Performed
<u>byteValue()</u>	Returns a byte value corresponding to this Double Object
<u>compare()</u>	Compare two primitive double values for numerical equality. As it is a static method therefore it can be used without creating any object of Double.
<u>compareTo()</u>	Used to compare two Double objects for numerical equality and returns a value less than 0, 0, a value greater than 0 for less than, equal to, and greater than.
<u>doubleValue()</u>	Returns a double value corresponding to this Double Object.
<u>doubleToLongBits()</u>	Returns the IEEE 754 floating-point "double format" bit layout of the given double argument.
<u>doubleToRawLongBits()</u>	Returns the IEEE 754 floating-point "double format" bit layout of the given double argument. It differs from the previous method as it preserves the Nan values.
<u>equals()</u>	Compare the equality of two Double objects and returns true if both the objects contain same double value.
<u>floatValue()</u>	Returns a float value corresponding to this Double Object.
<u>hashCode()</u>	Returns the hashcode corresponding to this Double Object.
<u>isInfinite()</u>	Returns true if the double object in consideration is very large, otherwise false.
<u>isNaN()</u>	Returns true if the double object in consideration is not a number, otherwise false.

Method	Action Performed
<u>intValue()</u>	Returns an integer value corresponding to this Double Object
<u>longValue()</u>	Returns long value corresponding to this Double Object.
longBitsToDouble()	Returns double value corresponding to the long bit pattern of the argument.
<u>parseDouble()</u>	Returns double value by parsing the string.
<u>shortValue()</u>	Returns short value corresponding to this Double Object
toHexString()	Returns hexadecimal representation of the argument double value.
toString()	Returns the string corresponding to the double value
valueOf()	Returns a Double-object initialized with the value provided
valueOf(String s)	Returns a Double-object initialized with the value provided

Implementation:

Java

```
// Java Program to Illustrate Double Class
// Via Demonstrating Its Methods

// Class
public class GFG {

    // Main driver method
    public static void main(String[] args)
    {

        // Declaring and initializing
        // double and String values
        double d = 12.34;
        String s = "GFG";
```

```

// Construct two Double objects
Double x = new Double(b);
Double y = new Double(bb);

// Method - toString()
System.out.println("toString(b) = "
                   + Double.toString(b));

// Method - valueOf()
// Return Double object
Double z = Double.valueOf(b);
System.out.println("valueOf(b) = " + z);
z = Double.valueOf(bb);
System.out.println("ValueOf(bb) = " + z);

// Method - parseDouble()
// Return primitive double value
double zz = Double.parseDouble(bb);
System.out.println("parseDouble(bb) = " + zz);

// Print statements
System.out.println("bytevalue(x) = "
                   + x.byteValue());
System.out.println("shortvalue(x) = "
                   + x.shortValue());
System.out.println("intvalue(x) = " + x.intValue());
System.out.println("longvalue(x) = "
                   + x.longValue());
System.out.println("doublevalue(x) = "
                   + x.doubleValue());
System.out.println("floatvalue(x) = "
                   + x.floatValue());

int hash = x.hashCode();
System.out.println("hashCode(x) = " + hash);

boolean eq = x.equals(y);
System.out.println("x.equals(y) = " + eq);

int e = Double.compare(x, y);
System.out.println("compare(x,y) = " + e);

int f = x.compareTo(y);
System.out.println("x.compareTo(y) = " + f);

Double d = Double.valueOf("1010.54789654123654");
System.out.println("isNaN(d) = " + d.isNaN());

System.out.println("Double.isNaN(45.12452) = "
                   + Double.isNaN(45.12452));

```

```

d = Double.valueOf(Double.POSITIVE_INFINITY + 1);
System.out.println(
    "Double.isInfinite(d) = "
    + Double.isInfinite(d.doubleValue()));

double dd = 10245.21452;
System.out.println("Double.toString(dd) = "
    + Double.toHexString(dd));

long double_to_long = Double.doubleToLongBits(dd);
System.out.println("Double.doubleToLongBits(dd) = "
    + double_to_long);

double long_to_double
    = Double.longBitsToDouble(double_to_long);
System.out.println(
    "Double.LongBitsToDouble(double_to_long) = "
    + long_to_double);
}
}

```

Output

```

toString(b) = 55.05
valueOf(b) = 55.05
ValueOf(bb) = 45.0
parseDouble(bb) = 45.0
bytevalue(x) = 55
shortvalue(x) = 55
intValue(x) = 55
longvalue(x) = 55
doublevalue(x) = 55.05
floatvalue(x) = 55.05
hashCode(x) = 640540672
x.equals(y) = false
compare(x,y) = 1
x.compareTo(y) = 1
isNaN(d) = false
Double.isNaN(45.12452) = false
Double.isInfinite(d) = true
Double.toString(dd) = 0x1.4029b7564302bp13
Double.doubleToLongBits(dd) = 4666857980575363115
Double.LongBitsToDouble(double_to_long) = 10245.21452

```

This article is contributed by **Rishabh Mahrsee**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

2

Related Articles

1. [Java.lang.Class class in Java | Set 2](#)

2. [Java.lang.Class class in Java | Set 1](#)

3. [Implement Triplet Class with Pair Class in Java using JavaTuples](#)

4. [Implement Quintet Class with Quartet Class in Java using JavaTuples](#)

5. [Implement Quartet Class with Triplet Class in Java using JavaTuples](#)

6. [Implement Octet Class from Septet Class in Java using JavaTuples](#)

7. [Implement Ennead Class from Octet Class in Java using JavaTuples](#)

8. [Implement Sextet Class from Quintet Class in Java using JavaTuples](#)

9. [Implement Septet Class from Sextet Class in Java using JavaTuples](#)

10. [Implement Decade Class from Ennead Class in Java using JavaTuples](#)

[Previous](#)

[Next](#)

Article Contributed By :



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Vote for difficulty

Current difficulty : [Basic](#)

Easy

Normal

Medium

Hard

Expert

Improved By : [rajeev0719singh](#), [abhishek0719kadiyan](#), [sagar0719kumar](#), [solankimayank](#), [bipin_kumar](#)

Article Tags : [Java-Classes](#), [Java-lang package](#), [java-wrapper-class](#), [Java](#)

Practice Tags : [Java](#)

[Improve Article](#)

[Report Issue](#)



A-143, 9th Floor, Sovereign Corporate Tower,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

[About Us](#)

[Careers](#)

[In Media](#)

[Contact Us](#)

[Privacy Policy](#)

[Copyright Policy](#)

[Third-Party Copyright Notices](#)

[Advertise with us](#)

Courses @Sale

[DSA \(Java/C++/Python/JS\)](#)

[Complete Interview Preparation](#)

[Full Stack Live](#)

[Data Science & ML](#)

[Android App Development](#)

[System Design Live](#)

Languages

[Python](#)

[Java](#)

Data Structures

[Array](#)

[String](#)

[GoLang](#)
[SQL](#)
[R Language](#)
[Android Tutorial](#)

[Stack](#)
[Queue](#)
[Tree](#)
[Graph](#)

Algorithms

[Sorting](#)
[Searching](#)
[Greedy](#)
[Dynamic Programming](#)
[Pattern Searching](#)
[Recursion](#)
[Backtracking](#)

Web Development

[HTML](#)
[CSS](#)
[JavaScript](#)
[Bootstrap](#)
[ReactJS](#)
[AngularJS](#)
[NodeJS](#)

Computer Science

[GATE CS Notes](#)
[Operating Systems](#)
[Computer Network](#)
[Database Management System](#)
[Software Engineering](#)
[Digital Logic Design](#)
[Engineering Maths](#)

Data Science & ML

[Data Science With Python](#)
[Data Science For Beginner](#)
[Machine Learning Tutorial](#)
[Maths For Machine Learning](#)
[Pandas Tutorial](#)
[NumPy Tutorial](#)
[NLP Tutorial](#)

Interview Corner

[Company Preparation](#)
[Preparation for SDE](#)
[Company Interview Corner](#)
[Experienced Interview](#)
[Internship Interview](#)
[Competitive Programming](#)
[Aptitude](#)

Python

[Python Tutorial](#)
[Python Programming Examples](#)
[Django Tutorial](#)
[Python Projects](#)
[Python Tkinter](#)
[OpenCV Python Tutorial](#)

GfG School

Write & Earn

CBSE Notes for Class 10

Pick Topics to Write

CBSE Notes for Class 11

Write Interview Experience

CBSE Notes for Class 12

Internships

English Grammar

Video Internship

@geeksforgeeks , Some rights reserved