1. What do Java Wrapper classes do?

Ans: Wrapper classes are predefined classes available in the java.lang package. We need Wrapper classes for converting primitive data types into object type data .

Java.lang package is by default add by jvm in every program.

1. List the eight primitive data types in Java.

Ans:

* boolean
* char
* byte
* short
* int
* long
* float
* double

1. Why do we need Wrapper classes?

Ans: we need Wrapper classes for converting primitive ­­­­data types into object.

1. What is autoboxing?

Ans: In Autoboxing we convert Primitive data types into Objects.

1. What is unboxing?

Ans: In Unboxing we convert Objects type data into Primitive data types.

1. Explain the purpose of the valueOf() method in Wrapper classes.

Ans: To convert Primitive data types into Objects.

1. Differentiate between == and equals() when comparing Wrapper objects.

Ans: The == operator compares the memory addresses (references) of objects, while the equals() method compares the actual content or state of objects.

1. Can you explain the significance of the hashCode() method in Wrapper classes?

The hashCode() method returns the hash code value for the object, which is crucial for hashing-based data structures like HashMap.

1. How can you convert a Wrapper object to a primitive data type?

Ans: There is a concept of Unboxing where we can convert Wrapper object into primitive data types.

1. Discuss the parseInt() method in the Integer class.

The parseInt() method is used to convert a String to an int primitive.

1. String numStr = "123";
2. **int** num = Integer.parseInt(numStr);

11. Explain the compareTo() method in the Comparable interface.

The compareTo() method is used to compare two Wrapper objects and returns a negative, zero, or positive value based on their order.

1. Integer a = 10;
2. Integer b = 5;
3. System.out.println(a.compareTo(b)); // 1 (a > b)
   1. How do you check if a given string is a valid representation of a particular primitive type?

Ans: To determine whether a string is a valid representation of a specified numeric type, use the static TryParse method that is implemented by all primitive numeric types and also by types such as DateTime and IPAddress.

* 1. Explain the purpose of the Boolean class in Java.

Ans: The Boolean class wraps a value of the primitive type boolean in an object.

* 1. How can you convert a boolean primitive to a Boolean object?

Ans: We can convert a boolean primitive to a Boolean object by using Boolean Class.

ADVERTISEMENT

* 1. Explain the toString() method in Wrapper classes.

The toString() method returns a string representation of the object, which is useful for printing or logging.

ADVERTISEMENT

1. Integer num = 42;
2. System.out.println(num.toString()); // "42"

20. How can you convert a Wrapper object to a String?

We can use the toString() method or simply concatenate it with an empty string.

1. Integer num = 42;
2. String numStr = num.toString();
3. // or
4. String numStr2 = num + "";

21. Explain the toBinaryString() method in the Integer class.

The toBinaryString() method converts an int to a binary string representation.

1. **int** num = 42;
2. String binaryStr = Integer.toBinaryString(num);

22. What is the purpose of the MAX\_VALUE and MIN\_VALUE constants in Wrapper classes?

ADVERTISEMENT

ADVERTISEMENT

These constants represent the maximum and minimum values of the primitive data types.

1. System.out.println(Integer.MAX\_VALUE); // 2147483647
2. System.out.println(Integer.MIN\_VALUE); // -2147483648

23. Discuss the parseXxx() methods in Wrapper classes.

The parseXxx() methods (e.g., parseInt(), parseDouble()) are used to convert a String to the corresponding primitive type.

1. String numStr = "16";
2. **int** num = Integer.parseInt(numStr);

ADVERTISEMENT

24. How can you create a BigInteger object from a String?

Ans: Use BigInteger create a BigInteger object from a String.

26. Explain the valueOf() method in the BigDecimal class.

Ans: ValueOf(long unscaledVal, int scale) is an inbuilt method in Java that is used to translate a long unscaled value and an int scale into a BigDecimal.

28. Can you use Wrapper classes in a switch statement?

Ans: Java Wrapper Classes in Switch Statements:-

Java provides four wrapper classes to use: Integer, Short, Byte, and Long in switch statements.

ADVERTISEMENT

ADVERTISEMENT