**198. Object Class.**

1. java.lang. package gets automatically imported into java program.

2. **Object class is a parent for all the classes of java.**

3. Every class in java is directly or indirectly inheriting from object class. Even user define classes inherits from object class.

4. Following are the methods in Object Class:

|  |  |
| --- | --- |
| protected [**Object**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html) | [**clone**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#clone--)()  Creates and returns a copy of this object. |
| boolean | [**equals**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#equals-java.lang.Object-)([**Object**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html) obj)  Indicates whether some other object is "equal to" this one. |
| protected void | [**finalize**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#finalize--)()  Called by the garbage collector on an object when garbage collection determines that there are no more references to the object. |
| [**Class**](https://docs.oracle.com/javase/8/docs/api/java/lang/Class.html)<?> | [**getClass**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#getClass--)()  Returns the runtime class of this Object. |
| int | [**hashCode**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#hashCode--)()  Returns a hash code value for the object. |
| void | [**notify**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#notify--)()  Wakes up a single thread that is waiting on this object's monitor. |
| void | [**notifyAll**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#notifyAll--)()  Wakes up all threads that are waiting on this object's monitor. |
| [**String**](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) | [**toString**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#toString--)()  Returns a string representation of the object. |
| void | [**wait**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#wait--)()  Causes the current thread to wait until another thread invokes the [**notify()**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#notify--) method or the [**notifyAll()**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#notifyAll--) method for this object. |
| void | [**wait**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#wait-long-)(long timeout)  Causes the current thread to wait until either another thread invokes the [**notify()**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#notify--) method or the [**notifyAll()**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#notifyAll--) method for this object, or a specified amount of time has elapsed. |
| void | [**wait**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#wait-long-int-)(long timeout, int nanos)  Causes the current thread to wait until another thread invokes the [**notify()**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#notify--) method or the [**notifyAll()**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#notifyAll--) method for this object, or some other thread interrupts the current thread, or a certain amount of real time has elapsed. |

5. Multithreading is supported by java by its core.

6. As methods of multithreading are present inside the object class.

7. hascode is a unique code assigned to every java object. Java supports hashing from its core.

8. We can print the object using toString() method. toString() method is called automatically when object is passed to System.out.println() method.

9.We can get hash code using hasCode() method.

10. We can also define our own hash code.

11. Every class in java inherits the Object class directly and indirectly.

12. You can override the hasCode(), toString(), isEqual and other methods.

13. We cannot override wait(), notify() and other similar methods because these methods are final.

**198. Wrapper Classes.**

1. As java is object-oriented programming language, therefore in java everything must be an object. And for every object there has to be class.

2. primitives are not objects. Primitives are built-in datatypes.

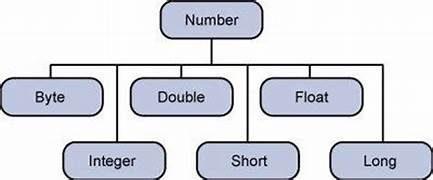
3. To treat java as Object Oriented Programming class we want classes for primitives also.

4. Java provides wrapper class. It is a wrapper around the primitives so they can become the object.

5. All the wrapper class are present in java.lang. package.

6. For every primitive datatype there is wrapper class.

7.



8. Number class has various methods which can use for datatype conversion.

9. We can assign value to Integer object by using valueOf() method. Or we can directly assign it value like we assigns it to primitive datatypes.

10. We can even convert the string into the value.

11. Boxing is process in which we wrap primitive in a class.

12. We can assign values from Wrapper class to primitive datatypes directly or by using datatypeValueu() method.

13. Process assigning value from Wrapper class object to primitive datatype is called as unboxing.

14.