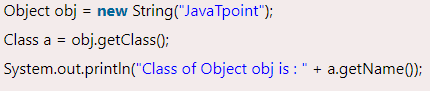
**Object class:**

Parent class of all the classes in java.

It is beneficial if you want to create and object whose type you done know.

Object class provides 11 methods:

getClass() : returns the runtime class of this object.



hashCode() : returns hashcode number of the object.

obj1.equals(Object obj2) : compares the given object to this object.

protected Object clone() throws CloneNotSupportedException : creates and returns the exact copy (clone) of this object.

toString() : string representation of the object.

void notify() : wakes up single thread, waiting on this object's monitor.

void notifyAll() : wakes up all the threads.

wait(long timeout,int nanos): causes the current thread to wait for the specified milliseconds, until another thread notifies.

finailize(): is invoked by the garbage collector before object is being garbage collected.

|  |  |
| --- | --- |
| **Shallow Copy** | **Deep Copy** |
| It is fast as no new memory is allocated. | It is slow as new memory is allocated. |
| Changes in one entity is reflected in other entity. | Changes in one entity are not reflected in changes in another identity. |
| The default version of the clone() method supports shallow copy. | In order to make the clone() method support the deep copy, one has to override the clone() method. |
| A shallow copy is less expensive. | Deep copy is highly expensive. |
| Cloned object and the original object are not disjoint. | Cloned object and the original object are disjoint. |

|  |  |  |
| --- | --- | --- |
| **JDK** | **JRE** | **JVM** |
| JDK is a software development kit to develop applications in Java. Contains tools for developing, debugging, and monitoring java code. | It is a software bundle which provides Java class libraries with necessary components to run Java code. | JVM executes Java byte code and provides an environment(JRE) for executing it. |
| JDK is platform dependent. | JRE is also platform dependent. | JVM is highly platform dependent. |
| JDK = Java Runtime Environment (JRE) + Development tools | JRE = Java Virtual Machine (JVM) + Libraries to run the application | JVM = Only Runtime environment for executing the Java byte code. |

Public : accessible within same class, outside the class, same package, outside package.

Protected : accessible within same class, same package, outside package by subclass(child) only.

Default : within the same package only.

Private : within same class only.

Why we can’t override static method ??

We can’t override the static method as it belongs to the class area. But a normal method inside a class belongs to an object(heap area) hence can be overridden.

Composition ??

Covariant return type ??

String immutable ??

Because if we try to contact and do any change , it won’t change.

Even if we rename a string normally, I creates another object in the string pool.

objects immutable ??

How many ways can we create the string object?

By String literal (double quotes)

String s="Welcome";

By new keyword

String s=**new** String("Welcome");

Here two objects are created, one in heap and other in string pool constant.

String is immutable whereas StringBuffer and StringBuilder are mutable classes.

StringBuffer is thread-safe and synchronized whereas StringBuilder is not. That’s why StringBuilder is faster than StringBuffer.

String concatenation operator (+) internally uses StringBuffer or StringBuilder class.

