-----Object Class Methods-----

to String():

Employee e = new employee ();

If try to call employee e --ref, automatically it call to String () method and print the address of the employee object stored in heap.

(It convert the has code of the address in to hexadecimal format and also adds package and class name)

This method is often overridden in user-defined classes to provide a meaningful string

Ex: Pojo classes like Person, Passport etc.

Boolean equals (Object ob.):

Indicates whether some other object is "equal to" this one. The default implementation in the Object class compares memory addresses.

****Object class equals method is meant for reference comparison

***We can also override equals method in user defined classes for content comparison

*** Predefined class like String override the equal's method and meant for content comparison only.

** All wrapper (Integer, Double, etc.) classes override the equals to provide their own implementation.

== Equals operator always meant for ref comparison, either in user defined or predefined classes.

Hash Code ():

Returns a hash code value for the object. The default implementation returns the memory address,

User defined class often overridden for better performance in hash-based data structures.

Get Class ():

Returns the runtime class of the object.

Clone ();

Clone () method is used to create a copy of an object. Clone method present in Object class.

To clone any class, class must implement the Cloneable interface, and the clone () method should be declared as public

Shallow Cloning and Deep Cloning

By default clone() method creates a shallow copy,(Ex: employee if class contains the properties like string double etc. it works fine with clone, In case employee contains any other Object Property like dept. as a ref clone

method fails to clone the department property than we required the deep cloning. To provide the deep cloning we need to customize the clone method)

If you need a deep copy, you may need to implement a custom clone () method to handle it.

clone able interface is a marker interface and JVM provided the implementation for clone method. When it observed class implemented clone able interface else through clone Not supported Exception

******

Wait

Notify()

NotifyAll()

Wait notify and notify all() methods are Object class method.

Two thread can communicate with each other through wait and notify methods.

Notify method: give the notification to only one waiting thread.

Notify All Method : Gives the notification to all waiting threads.

Once thread get the notification on waiting thread, Thread moves from waiting state to Runnable state.

If waiting thread not received any notification from any other thread than it remains in waiting state only.

Wait (Nano seconds) : Thread remains in waiting state for that particular time only, In case of no notification received it automatically comes out of waiting state to runnable.

Finalize ():

The finalize () method is declared as protected in the Object class and throws the Throw able class.

Protected void finalize () **throws** Throw able { }

The finalize () method is automatically called by the garbage collector It removes un reference Object from heap memory. It is not called explicitly by the developer. (If the developer want to call the garbage collector we need to call System.gc ())

finalize () is called is not guaranteed, and it depends on the garbage collector. Therefore, relying on finalize () for critical resource cleanup or state management is not recommended.

Below are the ways Object eligible for garbage collector

Null Reference

Employee e = null;

When Object referred to null than Object is eligible for garbage collector.

Reassigning Reference Variables:

Employee e1 = new employee ();

Employee e2 = new employee ();

e1 == e2

Object is eligible for garbage collector.

Object Goes Out of Scope:

If an object is created within a method, and the method completes its execution, the object goes out of scope, making it eligible for garbage collection.

Anonymous class

Anonymous class is a class that is defined without a name. It is typically used for creating a one-time-use class for a specific purpose

Ex: new employee ();