**java.lang package – Part-04**

* **getClass() Method:**

We can use getClass() method to get runtime class definition of an object.

public final Class getClass();

By using this Class class object we can access class level properties like fully qualified name of the class, methods information, constructors information etc.

Example:

import java.lang.reflect.\*;

class Test{

public static void main(String[] args){

int count = 0;

Object o = new String(“Saravana”);

Class c = o.getClass();

System.out.println(“Fully Qualified Name:”+c.getClassName());

Method[] m = c.getDeclaredMethods();

for(Method m1: m){

System.out.println(m1.getName());

Count++;

}

System.out.println(“Total count:”+count);

}

}

**Example\_02: To display database vendor specific connection interface implemented class name.**

Connect con = DriverManager.getConnection(“”);

System.out.println(con.getClass().getName());

Note:

1. After loading every .class file JVM will create an Object of the type java.lang.Class in the heap area. Programmers can use this object to get class level information.
2. We can use getClass() method very frequently in reflections.

* **finalize() Method:**

Just before destroying an object, garbage collector calls finalize method to perform cleanup activities.

Once finalize method completes automatically garbage collector destroys that object.

* **wait(), notify(), notifyAll():**

We can use these methods for inter thread communication.

The thread which is expecting update, responsible to call wait(). Then immediately the thread will enter into waiting state.

The thread which is responsible to perform updation after performing updation, the thread can call notify method.

The waiting thread will get that notification and continue its execution with those updates.