Analysing/Examining the contents(data/method/superclass) of the class at runtime

3 ways to create Class object in Reflection API

Class c=Class.*forName*("Student");

Or

Student s1=**new** Student();

Class c2=s1.getClass();

Or

Class c3=Student.**class**;

**Now this Class Object will store the metadata(Internal Information) about the Student class**

NOTE :

JVM uses **“public/default” Zero Args** constructor to initialize the object,

If no args constructoris not declared in the class it will throw InstantiationException

If no args constructoris is made privart it will throw InstantiationException

So JVM should know what all methods are there in the class, for this Introspection JVM uses Reflection API present in java.lang.reflect

java.lang.Class - > Gather the entire meta data of a class

java.lang.reflect.Field - > Gather the entire details about the class Data member i.e Access modifiers, var name, value, type, datatype

java.lang.reflect.Method - > Gather the entire details about the class member method i.e Access modifier, method name, return type, parameter list

java.lang.reflect.Constructor- > Gather the entire details about the class Constructors i.e Access modifier, parameter list

java.lang.reflect.Modifier - > Gather the entire details about the Modifier

int i=c.getModifiers();

System.out.println(Modifier.toString(i));