Reflection API:

Reflection is the process of examining or modifying the runtime behaviour of a class at runtime. The java.lang.**Class** class provides many methods that can be used to get **metadata**, examine and change the runtime behaviour of a class.

**Where is it used?**

The Reflection API is mainly used in: IDE (Integrated Development Environment) e.g. Eclipse, MyEclipse, NetBeans etc.

Debugger

Test Tools etc.

**java.lang.Class class:**

The java.lang.Class class performs mainly two tasks:

Provides methods to get the metadata of a class at runtime.

Provides methods to examine and change the runtime behavior of a class.

**Commonly used methods of Class class:**

1) public String getName()

2) public static Class forName(String className)throws ClassNotFoundException

3) public Object newInstance()throws InstantiationException,IllegalAccessException

4) public boolean isInterface()

5) public boolean isArray()

6) public boolean isPrimitive()

7) public Class getSuperclass()

8) public Field[] getDeclaredFields()throws SecurityException

9) public Method[] getDeclaredMethods()throws SecurityException

10) public Constructor[] getDeclaredConstructors()throws SecurityException

11) public Method getDeclaredMethod(String name,Class[] parameterTypes)throws NoSuchMethodException,SecurityException

**How to get the object of Class class?**

forName() method of Class class

getClass() method of Object class

the .class syntax

**1) forName() method of Class class:**

is used to load the class dynamically.

returns the instance of Class class.

It should be used if you know the fully qualified name of class. This cannot be used for primitive types.

**Example of forName() method:**

**class Simple{}**

**class Test{**

**public static void main(String args[]){**

**Class c=Class.forName("Simple");**

**System.out.println(c.getName());**

**}**

**}**

**getClass() method of Object class**

**class Simple{}**

**class Test{**

**void printName(Object obj){**

**Class c=obj.getClass();**

**System.out.println(c.getName());**

**}**

**public static void main(String args[]){**

**Simple s=new Simple();**

**Test t=new Test();**

**t.printName(s);**

**}**

**}**

**The .class syntax**

**Example of .class syntax:**

**class Test{**

**public static void main(String args[]){**

**Class c = boolean.class;**

**System.out.println(c.getName());**

**Class c2 = Test.class;**

**System.out.println(c2.getName());**

**}**

**}**