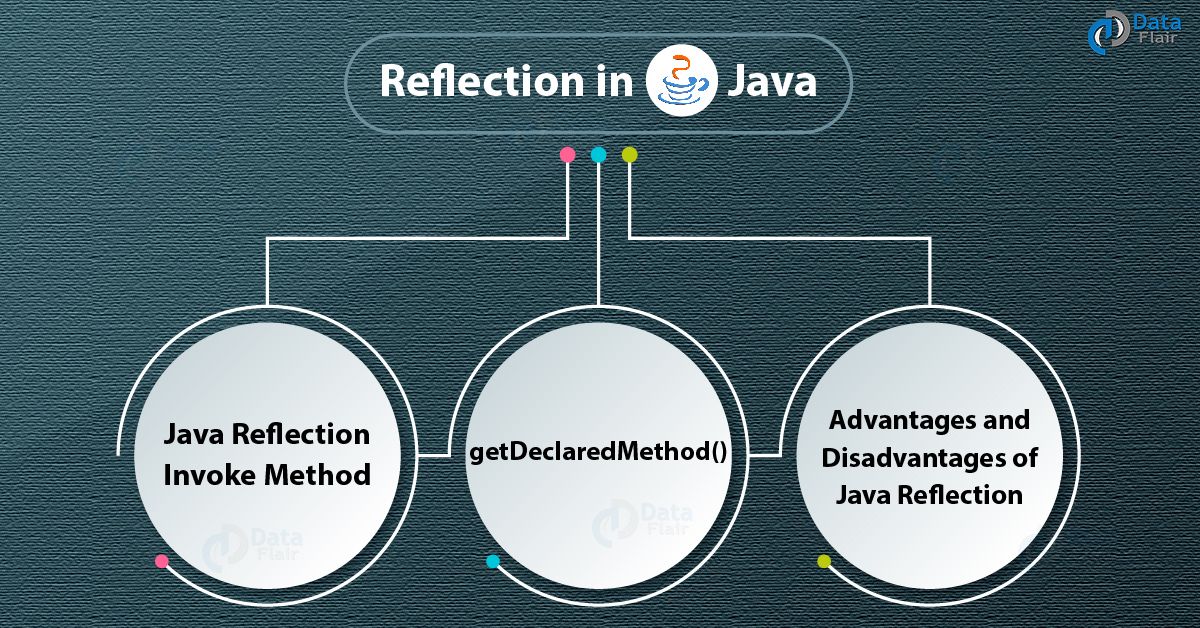
## ***Reflection in Java***

Reflection is a runtime API for inspecting and changing the behavior of methods, classes, and interfaces. The java.lang.reflect package contains the required reflection classes. Reflection tells about the class to which an object belongs and the methods of that class that can be used for the object. One can call the methods at runtime using reflection regardless of the access specifier used.

Reflection helps when thinking about:

* ****Class****: The getClass() function is used to figure out what class an object belongs to.
* ****Constructors****: Reflection helps to get the public constructors of the class to which an object belongs using the getConstructors() function.
* ****Methods****: The public methods of the class to which an object belongs are obtained using the getMethods() method.



**Example:**

class Students{

    String name;

    int roll\_no, grade;

    public void setStudent(String n, int r, int c){

        name = n;

        roll\_no = r;

        grade = c;

}

    public void getStudent(){

        System.out.println("Student Name = " + name);

        System.out.println("Roll No = " + roll\_no);

        System.out.println("Class = " + grade);

    }

}

class Teachers{

    String name;

    double salary;

    int contact\_no;

    public void setTeacher(String n, int s, int c){

        name = n;

        salary = s;

        contact\_no = c;

}

    public void getTeacher(){

        System.out.println("Teacher Name = " + name);

        System.out.println("Salary = " + salary);

        System.out.println("Contact Number = " + contact\_no);

    }

}

public class School{

    public static void main(String[] args)

    {

        Students students = new Students();

        students.setStudent("Sujit Khanal", 10, 14);

        students.getStudent();

        Teachers teachers = new Teachers();

        teachers.setTeacher("Teacher Sir", 10000, 984123456);

        teachers.getTeacher();

    }

}

## **Uses:**

The Reflection API is mostly used in the following applications:

* Eclipse, MyEclipse, NetBeans, and other IDEs (Integrated Development Environments) are examples.
* Debugging
* Tools for testing, etc.

## **The Benefits of Reflection:**

* ****Extensibility Features:**** By generating instances of extensibility objects with their fully-qualified names, an implementation may use external, user-defined classes.
* ****Debugging and debugging tools:**** Debuggers inspect private members on classes using the reflection property.