

Task 1: Reflection API

Use reflection to inspect a class's methods, fields, and constructors, and modify the access level of a private field, setting its value during runtime

ANS:

```
package com.Day24;
import java.lang.reflect.*;
public class ReflectionDemo {
    public static void main(String[] args) {
        try {
            // Load the Person class
            Class<?> personClass = Class.forName("com.Day24.Person");
            // Inspect constructors
            System.out.println("Constructors:");
            Constructor<?>[] constructors = personClass.getDeclaredConstructors();
            for (Constructor<?> constructor : constructors) {
                System.out.println(constructor);
            }
            // Inspect fields
            System.out.println("\nFields:");
            Field[] fields = personClass.getDeclaredFields();
            for (Field field : fields) {
                System.out.println(field);
            }
            // Inspect methods
            System.out.println("\nMethods:");
            Method[] methods = personClass.getDeclaredMethods();
            for (Method method : methods) {
                System.out.println(method);
            }
            // Modify access level of the private field 'name' and set its value
            Person personInstance = new Person("John Doe");
            Field nameField = personClass.getDeclaredField("name");
            nameField.setAccessible(true);
            nameField.set(personInstance, "Jane Doe");
            // Verify the value change
            System.out.println("\nModified name field value: " + personInstance.getName());
            // Invoke the private method
            Method printNameMethod = personClass.getDeclaredMethod("printName");
            printNameMethod.setAccessible(true);
            printNameMethod.invoke(personInstance);
        } catch (ClassNotFoundException | NoSuchFieldException | IllegalAccessException |
            NoSuchMethodException | InvocationTargetException e) {
            e.printStackTrace();
        }
    }
}
```

```
package com.Day24;
public class Person {
    private String name;
```

```
public Person(String name) {  
    this.name = name;  
}  
public String getName() {  
    return name;  
}  
private void printName() {  
    System.out.println("Name: " + name);  
}  
}
```