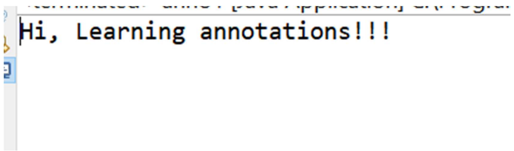
Annotations

1

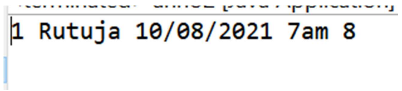
```java
package annotations;

import java.lang.annotation.ElementType;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;
import java.lang.annotation.Retention;

public class anno1 {

	public static void main(String[] args) {
		message a= new message("Hi, Learning annotations!!!");
		System.out.println(a.getName());

	}

}

@Target(ElementType.METHOD)
@Retention(RetentionPolicy.RUNTIME)
@interface Test{

}

class message
{
	String name;

	public message (String name)
	{
		this.name=name;
	}
	@Test
	String getName() {
		return name;
	}

}
```



```
Hi, Learning annotations!!!
```

2

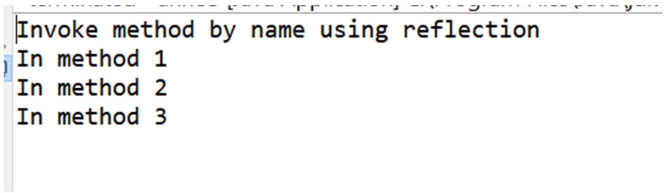
```java
package annotations;

import java.lang.annotation.ElementType;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;
import java.lang.annotation.Retention;

@Target(ElementType.TYPE)
@Retention(RetentionPolicy.RUNTIME)
@interface Info
{
        int DevId();
        String Devname();
        String date();
        String time();
        int version();
        }

@Info(DevId=1, Devname="Rutuja", date="10/08/2021", time="7am", version= 8)
class message1 {
        String name;

        public message1 (String name) {
                super();
                this.name = name;
        }

        public String getName() {
                return name;
        }
}

public class anno2 {

        public static void main(String[] args) {
                message1 obj = new message1("Hi");
                Class c=obj.getClass();
                java.lang.annotation.Annotation an= c.getAnnotation(Info.class);
                Info s=(Info)an;
                System.out.println(s.DevId()+" "+s.Devname()+" "+s.date()+"
"+s.time()+" "+s.version());

        }

}
```



```
1 Rutuja 10/08/2021 7am 8
```

3

```
package annotations;

import java.lang.annotation.ElementType;
import java.lang.annotation.RetentionPolicy;
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
import java.lang.annotation.Target;
import java.lang.annotation.Retention;

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)

@interface Execute{
        int sequence();
}

class anno3 {
        @Execute(sequence=1)
        public void MyMethod1()
        {
                System.out.println("In method 1");
        }
        @Execute(sequence=2)
        public void MyMethod2()
        {
                System.out.println("In method 2");
        }
        @Execute(sequence=3)
        public void MyMethod3()
        {
                System.out.println("In method 3");
        }



        public static void main(String[] args) throws Exception {
                System.out.println("Invoke method by name using reflection");

                anno3 obj =new anno3();
                Class<?> classObj= obj.getClass();

                Method MyMethod1= classObj.getDeclaredMethod("MyMethod1");
                Method MyMethod2= classObj.getDeclaredMethod("MyMethod2");
                Method MyMethod3= classObj.getDeclaredMethod("MyMethod3");

                try {
                        MyMethod1.invoke(obj);
                        MyMethod2.invoke(obj);
                        MyMethod3.invoke(obj);

                }

                catch(InvocationTargetException e)
                {
                        System.out.println(e.getCause());
                }
        }

}
```



```
Invoke method by name using reflection
In method 1
In method 2
In method 3
```