1.ANNOTATION PROGRAM

```
package anno;
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;
import java.lang.annotation.*;
import java.lang.reflect.*;
@Target(ElementType.METHOD)
@Retention(RetentionPolicy.RUNTIME)

@interface Test{

        String TestCase();
}

class test1{
         @Test(TestCase = "This is just a test case for Annotation")
         public void sayHello()
           {
                 System.out.println("hello annotation");
                 }
        }
public class annaOne {
        public static void main(String[] args) throws Exception {
                test1 t = new test1();
                Method d = t.getClass().getMethod("sayHello");
                Annotation ann = d.getAnnotation(Test.class);
                Test ts = (Test)ann;
System.out.println( ts.TestCase());
}}
```

2.program

```
package anno;

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



@Target(ElementType.TYPE)
@Retention (RetentionPolicy.RUNTIME)
@interface Info  {

        int AuthorID () default 1;
                String Author()  default "Glenn";
                String Supervisor()  default "raj";
                String Date() default "01/11/2021";
                String TIme() default "11:11";
                int Version() default 10;
                String Description() default "This is just an assignment on annotations";

    }


@Info ()
class myclass{



    }
public class annaTwo {
```

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

                        myclass ns = new myclass();



                        Class f  = ns.getClass();
                        Annotation annn = f.getAnnotation(Info.class);
                        Info i = (Info)annn;
                        System.out.println("AuthorID:"+i.AuthorID());
                        System.out.println("Author:"+i.Author());
                        System.out.println("Supervisor:"+i.Supervisor());
                        System.out.println("Date:"+i.Date());
                        System.out.println("Time:"+i.TIme());
                        System.out.println("Version:"+i.Version());
                        System.out.println("Description:"+i.Description());



                }

        }
```

3.program annotations

```
package anno;

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



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

```java
@interface Execute{

	int Sequence() ;
}


class MyClass1{

	  @Execute(Sequence=2)
	  public void myMethod1()
	      {
	        System.out.println("In method1 giving priority to sequence 2");
	        }

	  @Execute(Sequence=1)
	  public void myMethod2()
	      {
	        System.out.println("In method2 giving priority to sequence 1");
	        }

	  @Execute(Sequence=3)
	  public void myMetho31()
	      {
	        System.out.println("In method3 giving priority to sequence 3");
	        }
	  }



public class annaThree {

	public static void main(String[] args) throws Exception {

		MyClass1 mc = new MyClass1();


		Method e = mc.getClass().getMethod("myMethod1");
		Annotation ann = e.getAnnotation(Execute.class);
		Execute ts = (Execute)ann;
		System.out.println( ts.Sequence());


	}

}
```