**ASSIGNMENT ON ANNOTATION** :

1.

**package** Annotation\_Assignment;

**import** java.lang.reflect.Method;

**import** java.lang.annotation.\*;

**import** java.lang.reflect.\*;

@Retention (RetentionPolicy.***RUNTIME***)

@Target(ElementType.***METHOD***)

**@interface** test

{

String name();

}

**class** demo

{

@test (name="Palak")

**public** **void** show()

{

}

}

**public** **class** first

{

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

{

demo d=**new** demo();

Method m =d.getClass().getMethod("show");

test t=m.getAnnotation(test.**class**);

System.***out***.println(t.name());

}

}

Out put:

Palak

2.

**package** Annotation\_Assignment;

**import** java.lang.annotation.Documented;

**import** java.lang.annotation.ElementType;

**import** java.lang.annotation.Retention;

**import** java.lang.annotation.RetentionPolicy;

**import** java.lang.reflect.Method;

@Documented

@Retention (RetentionPolicy.***RUNTIME***)

**@interface** info

{

**int** AuthorID();

String Author\_Name();

String SuperVisor();

String Date();

String Time();

**int** version();

}

//@info(AuthorID=2,Author\_Name="Palak" ,SuperVisor="Pragati" , Date="18/01/2022" , Time="09:57:00",version=18)

**class** Author

{

@info(AuthorID=2,Author\_Name="Palak" ,SuperVisor="Pragati" , Date="18/01/2022" , Time="09:57:00",version=18)

**public** **void** show()

{

System.***out***.println("Hello ! Welcome ");

}

}

**public** **class** Second {

**public** **static** **void** main(String[] args) **throws** NoSuchMethodException , SecurityException

{

Author A= **new** Author();

A.show();

Method m = A.getClass().getMethod("show");

info I= m.getAnnotation(info.**class**);

System.***out***.println(I.AuthorID());

System.***out***.println(I.Author\_Name());

System.***out***.println(I.SuperVisor());

System.***out***.println(I.Date());

System.***out***.println(I.Time());

System.***out***.println(I.version());

}

}

Output:

Hello ! Welcome

2

Palak

Pragati

18/01/2022

09:57:00

18

3.

**package** Annotation\_Assignment;

**import** java.lang.annotation.ElementType;

**import** java.lang.annotation.Retention;

**import** java.lang.annotation.RetentionPolicy;

**import** java.lang.annotation.Target;

**import** java.lang.reflect.Method;

@Retention (RetentionPolicy.***RUNTIME***)

@Target(ElementType.***METHOD***)

**@interface** Execute

{

**int** Sequence();

}

**class** Sequence

{

@Execute(Sequence=3)

**public** **void** method\_first()

{

System.***out***.println("Show First method ");

}

@Execute(Sequence=1)

**public** **void** method\_second()

{

System.***out***.println("Show Second method ");

}

@Execute(Sequence=2)

**public** **void** method\_third()

{

System.***out***.println("Show third method ");

}

}

**public** **class** third {

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

{

Sequence seq= **new** Sequence();

Method[] method=seq.getClass().getMethods();

**for**(Method m : method)

{

Execute e = m.getAnnotation(Execute.**class**);

**if**(e!=**null**)

{

**try**

{

m.invoke(seq);

}

**catch**(Exception ex)

{

System.***out***.println(ex);

}

}

}

}

}