**Assignment 7**

**1]ANS:-**

import java.lang.annotation.\*;

import java.lang.reflect.\*;

@Retention (RetentionPolicy.RUNTIME)

@Target (ElementType.METHOD)

@interface Test

{

String str();

}

class First

{

@Test(str="Test Annotation")

public void testCase()

{

}

}

import java.lang.reflect.Method;

public class TestMainmethod {

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

First f=new First();

Method m = f.getClass().getMethod("testCase");

Test ts = m.getAnnotation(Test.class);

System.out.println(ts.str());

}

}

**OUTPUT:-**

Test Annotation

**2]ANS:-**

import java.lang.annotation.Documented;

import java.lang.annotation.Retention;

import java.lang.annotation.RetentionPolicy;

@Documented

@Retention (RetentionPolicy.RUNTIME)

@interface info

{

int id();

String name();

String superviser();

String date();

String time();

int version();

}

public class Author {

@info(id = 1206, name = "Rushikesh",

superviser ="Anand",date="12/05/2019",time="11:50:50 hrs",version=12)

public void display()

{

System.out.println("provide details");

System.out.println();

}

}

import java.lang.reflect.Method;

public class AuthorMainmethod {

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

Author a = new Author();

a.display();

Method m = a.getClass().getMethod("display");

info i = m.getAnnotation(info.class);

System.out.println("Author ID : " +i.id());

System.out.println("Author Name: " +i.name());

System.out.println("Superviser Name: " +i.superviser());

System.out.println("Date: " +i.date());

System.out.println("Time: " +i.time());

System.out.println("Version: " +i.version());

}

}

**OUTPUT:-**

provide details

Author ID : 1206

Author Name: Rushikesh

Superviser Name: Anand

Date: 12/05/2019

Time: 11:50:50 hrs

Version: 12

**3]ANS:-**

Class MyClass

{

@Execute(Sequence=2)

Public void myMethod1()

{

}

@Execute(Sequence=1)

Public void myMethod2()

{

}

@Execute(Sequence=3)

Public void myMethod3()

{

}

import java.lang.annotation.ElementType;

import java.lang.annotation.Retention;

import java.lang.annotation.RetentionPolicy;

import java.lang.annotation.Target;

@Target(value = ElementType.METHOD)

@Retention(RetentionPolicy.RUNTIME)

@interface Execute

{

int Sequence();

}

class Sequence

{

@Execute(Sequence=2)

public void method1()

{

System.out.println("Method 1");

}

@Execute(Sequence=1)

public void method2()

{

System.out.println("Method 2");

}

@Execute(Sequence=3)

public void method3()

{

System.out.println("Method 3");

}

}

import java.lang.reflect.Method;

public class SequenceMainmethod

{

public static void main(String[] args)

{

Sequence s = new Sequence();

Method[] methods = s.getClass().getMethods();

for (Method method : methods)

{

Execute exe = method.getAnnotation(Execute.class);

if (exe != null)

{

try

{

method.invoke(s);

} catch (Exception e)

{

e.printStackTrace();

}

}

}

}

}

**OUTPUT: -**

Method 2

Method 1

Method 3