

**SAP Id: 500091584**  
**Roll No: R2142210822**  
**Name: Ujjwal Kumar Gupta**

**Course: CSE DevOps (Hons.)**  
**Program & Sem: B.tech & 3rd**  
**Session : 2021-25**

## **INDEX**

<b>S.No.</b>	<b>Experiment No.</b>	<b>Title</b>	<b>Date of Performance</b>	<b>Date of Submission</b>	<b>Remarks (By Faculty)</b>

Note: 1. Submit your original work otherwise you will lose marks.

2. Submit all your experiments into a single file

3. The submission must be properly compiled and in provided format only

## Experiment No. 5

**Date of performance: 21/09/2022**

**Date of Submission: 21/09/2022**

**SAP Id: 500091584**

**Roll No.: R2142210822**

**Name of the Student: Ujjwal Kumar Gupta**

---

**1. Title:** Interface

**2. Objective:** Interface

**3. List of lab activities:**

- 1) Write a program to create interface named test. In this interface the member function is square. Implement this interface in arithmetic class. Create one new class called ToTestInt. In this class use the object of arithmetic class.
- 2) Write a program to create interface A, in this interface we have two methods meth1 and meth2. Implements this interface in another class named MyClass.
- 3) Write a program in Java to show the usefulness of Interfaces as a place to keep constant value of the program
- 4) Write a program to create an Interface having two methods division and modules. Create a class, which overrides these methods.

**4. Algorithm/Flowchart and Code followed by Output screenshot (2 samples for each program):**

The screenshot shows the Visual Studio Code editor with a file named `testint.java` open. The code defines an interface `test` with a method `square()`, and a class `arithmetic` that implements `test` by providing a `square()` method that prints the square of a given number. A `main` method in the `testint` class creates an instance of `arithmetic` and calls its `square()` method. The terminal at the bottom shows the command `cd "C:\Users\Ujjwal\Documents\starting\src\Experiment 4> cd "C:\Users\Ujjwal\Documents\starting\src\Experiment 5>` and the output `square of given number is : 100`.

```
src > Experiment 5 > J twomethod.java > ...
1 interface A
2 {
3     void meth1();
4     void meth2();
5 }
6 class MyClass implements A
7 {
8     public void meth1()
9     {
10         System.out.println("Method 1 running");
11     }
12     public void meth2()
13     {
14         System.out.println("Method 2 running");
15     }
16 }
17
18 public class twomethod {
19     Run | Debug
20     public static void main(String[] args) {
21         MyClass m1= new MyClass();
22         m1.meth1();
23         m1.meth2();
24     }
25 }
26
```

PS C:\Users\Ujjwal\Documents\starting\src\Experiment 5> cd  
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 5> cd "C:\Users\Ujjwal\Documents\starting\src\Experiment 5" & .\twomethod.exe  
Method 1 running  
Method 2 running  
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 5>

```
Run Terminal Help coneabst.java - starting - Visual Studio Code

J coneabst.java X

src > Experiment 5 > J coneabst.java > cylinder

1 interface shape {
2     static final float pi = 3.14f;
3
4     void area(float r, float h);
5
6     void volume(float r, float h);
7
8 }
9
10 class cylinder implements shape {
11     public void area(float r, float h) {
12         float ans = 2 * pi * r * r * (h + r);
13         System.out.println("TSA of cylinder : " + ans);
14     }
15
16     public void volume(float r, float h) {
17         float ans = pi * r * r * h;
18         System.out.println("Volume of cylinder : " + ans);
19     }
20 }
21
22
23 class cone implements shape {
24     public void area(float r, float l) {
25
26         float ans = pi * r * r * (l + r);
27         System.out.println("TSA of the cone : " + ans);
28     }
29
30 }
31
32 public void volume(float r, float h) {
33
34 }
35
36 }
37
38 public class coneabst {
39     Run | Debug
40     public static void main(String[] args) {
41         shape s1 = new cylinder();
42         s1.area(4.2f, 8.7f);
43         s1.volume(8.6f, 7.4f);
44         System.out.println("\n");
45         shape s2 = new cone();
46         s2.area(8.6f, 5.8f);
47         s2.volume(2.3f, 5.6f);
48     }
49 }
50
```

```

t }
TSA of cylinder : 348.2504
volume of cylinder : 1718.5348

TSA of the cone : 388.85764
volume of the cone : 31.006453
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 5>

```

```

src > Experiment 5 > J overrides.java > student > modules(int)
1  interface course
2  {
3      void division(int d);
4      void modules(int m);
5  }
6  class student implements course
7  {
8      String name;
9      int div, mod;
10     void name(String n)
11     {
12         name=n;
13     }
14     public void division(int d)
15     {
16         div=d;
17     }
18     public void modules(int m)
19     {
20         mod=m;
21     }
22     void display()
23     {
24         System.out.println("name of the student :"+ name);
25         System.out.println("Division :"+ div);
26         System.out.println("total modules :"+ mod);
27     }
28 }
29 public class overrides {
30     Run | Debug
31     public static void main(String[] args) {
32         student s1= new student();
33         s1.name(n: "ujjwal");
34         s1.division(d: 12);
35         s1.modules(m: 30);
36         s1.display();
37     }
38 }

```

```

OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
total modules :30
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 5> cd "C:\Users\Ujjwal\Documents\start
des"
name of the student :ujjwal
Division :12
total modules :30
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 5>

```

## 5. Brief notes about all the concepts related to the lab experiment

An interface in Java is a blueprint of a class. It has static constants and abstract methods.

The interface in Java is *a mechanism to achieve abstraction*.

```

interface <interface_name>{

    // declare constant fields
    // declare methods that abstract
    // by default.
}

```