

# Programming Paradigms Laboratory

B.Tech.



**Name** : Subhendu Maji

**Roll Number** : 18ETCS002121

**Department** : Computer Science and Engineering

**Faculty of Engineering & Technology**  
**Ramaiah University of Applied Sciences**

Faculty	Engineering & Technology
Programme	B. Tech. in Computer Science and Engineering
Year/Semester	2 <sup>nd</sup> Year / 4 <sup>th</sup> Semester
Name of the Laboratory	Programming Paradigms Laboratory
Laboratory Code	19CSL217A

## Laboratory 7

Title of the Laboratory Exercise: Abstract class and Packages

### 1. Questions

- a. Develop a Java program to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for each of the two classes and print the percentage of marks for both the students.
- b. Develop a program to identify the accessibility of a variable by means of different access specifiers within and outside package.

### 2. Calculations/Computations/Algorithms

```
package student;
import java.util.*;
abstract class marks{
    abstract void getpercentage();
}
class a extends marks{
    int a1,a2,a3;
    a(int a,int b,int c){
        a1=a;a2=b;a3=c;
    }
    void getpercentage(){
        int a=a1+a2+a3;
        double result=(a*100)/300;
        System.out.println("First set of student's percentage is "+result);
    }
}
class b extends marks
{
    int a4,a5,a6,a7;
    b(int d,int e,int f,int g){
        a4=d;a5=e;a6=f;a7=g;
    }
    void getpercentage(){
        int a=a4+a5+a6+a7;
        double result;
        result=(a*100)/400;
        System.out.println("Second set of student's percentage is "+result);
    }
}
```

```

public class Student {

    public static void main(String[] args) {
        int a,b,c,d,e,f,g;
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the first set of students marks");
        a=obj.nextInt();
        b=obj.nextInt();
        c=obj.nextInt();
        System.out.println("Enter the second set of students marks");
        d=obj.nextInt();
        e=obj.nextInt();
        f=obj.nextInt();
        g=obj.nextInt();
        a obj1=new a(a,b,c);
        b obj2=new b(d,e,f,g);
        obj1.getpercentage();
        obj2.getpercentage();
    }

}

```

Figure 1.1 shows using abstract class and abstract method to find the percentage of marks obtained

```

package access;
import access.access1.*;
public class Access extends access1 {
    private void ret()
    {
        System.out.print("HIMANSHU ");
    }

    public static void main(String[] args) {
        Access a = new Access();
        a.ret();
        a.call();
    }

}

```

```
package access;

class access1 {
    protected void call()
    {
        System.out.println("JAIN");
    }
}
```

Figure 2.1 shows to identify the accessibility of a variable by means of different access specifiers within and outside package.

### 3. Presentation of Results

```
Enter the first set of students marks
55
67
88
Enter the second set of students marks
65
95
56
78
First set of student's percentage is %70.0
Second set of student's percentage is %73.0
BUILD SUCCESSFUL (total time: 30 seconds)
```

Figure 1.2 output of the abstract class and method to find the percentage of students

```
BUILD SUCCESSFUL (total time: 1 second)
```

Figure 2.2 output of the accessibility of the variable

#### 4. Conclusions

Abstraction is a process of hiding the implementation details and showing only functionality of the user, it shows only essential things to the user and hides the internal details. It focuses on what's the object does instead of how it does it.

#### 5. Limitations of Experiments and Results

A class which is declared as abstract is known as an abstract class. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be instantiated.