Programming Paradigms Laboratory

B.Tech.

Name : KAUSHAL VASHISTH

Roll Number : 18ETCS002147

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	2nd Year / 4th Semester
Name of the Laboratory	Programming Paradigms Laboratory
Laboratory Code	19CSL217A

Name: KAUSHAL VASHISTH Register Number: 18ETCS002147

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

Part A:-

```
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("1st list  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("2nd list of student's percentage is %" +
 result);
    }
}
```

Name: KAUSHAL VASHISTH Register Number: 18ETCS002147

```
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("input the 1st list of scholar marks");
        a = obj.nextInt(); b = obj.nextInt();
        c = obj.nextInt();
        System.out.println("input the 2nd list  of scholar 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();
    }
}
```

Part B:-

```
package access;
public class Access extends access1 {
    private void ret(){
        System.out.println("gaurav");
    }
    public static void main(String[] args) {
        Access a =new Access();
        a.ret();
        a.call();
    }
}
```

```
package access;
public class access1 {
    protected void call(){
        System.out.println("gupta");
    }
}
```

3. Presentation of Results :-

Part A:-

```
run:
input the 1st list of scholar marks
78
98
99
input the 2nd list  of scholar marks
100
99
89
87
1st list  of student's percentage is %91.0
2nd list of student's percentage is %93.0
BUILD SUCCESSFUL (total time: 20 seconds)
```

Part B:-

```
run:
gaurav
gupta
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Conclusions :-

Abstract class in Java

A class which is declared with the abstract keyword is known as an abstract class in Java. It can have abstract and non-abstract methods (method with the body).

Before learning the Java abstract class, let's understand the abstraction in Java first.

Points to remember:-

- An abstract class must be declared with an abstract keyword.
- It can have abstract and non-abstract methods.
- It cannot be instantiated.
- It can have constructors and static methods also.
- It can have final methods which will force the subclass not to change the body of the method.

5. Limitations of Experiments and Results:-

Abstract classes cannot be instantiated; they must be subclassed, and actual implementations must be provided for the abstract methods. Any implementation specified can, of course, be overridden by additional subclasses. An object must have an implementation for all of its methods.