**Practical No. 18: Develop program for implementation of Single and Multilevel inheritance.**

1. **Practical Significance:**

Inheritance helps to reuse the existing class properties to derive a new class with additional properties. It helps to reduce the memory space, time, frustration and increases the reliability of code. Different types of Inheritance are used to extend the classes in different ways. The student will be able to use different types of inheritance.

1. **Relevant Course Outcome:**

Apply concept of inheritance for code reusability.

1. **Practical Outcome:**

Develop program for implementation of Single and Multilevel inheritance.

1. **Minimum Theoretical Background:**

**Inheritance**

The process of deriving a new class form an old class is called as Inheritance.

**Types of Inheritance:**

1. Single Inheritance
2. Multiple Inheritance
3. Hierarchical Inheritance
4. Multilevel Inheritance

Java does not directly implement multiple inheritance, is it implemented using a secondary inheritance path in the form of interfaces.

A

A

A

A

(c)Hierarchical Inheritance

A

B

(a)Single Inheritance

A

A

A

(b)Multiple Inheritance

A

B

C

(d)Multilevel Inheritance

**Defining a subclass:**

class subclassname extends superclassname

{

variables declaration;

method declaration;

}

1. **Program Code:**

**Single Inheritance**

class Person

{

String name = "Aniket";

public void show()

{

System.out.println("My name is :");

}

}

class Student extends Person

{

String course = "Java Programming";

public void show1()

{

System.out.println("I Love to learn :");

}

}

public class SingleInheritance

{

public static void main(String args[])

{

Student obj = new Student();

obj.show();

System.out.println(obj.name);

obj.show1();

System.out.println(obj.course);

}

}

**Multilevel Inheritance**

class A

{

public void disA()

{

System.out.println("ClassA");

}

}

class B extends A

{

public void disB()

{

System.out.println("ClassB");

}

}

class C extends B

{

public void disC()

{

System.out.println("ClassC");

}

}

public class Multilevel extends C

{

public static void main(String args[])

{

C c = new C();

c.disA();

c.disB();

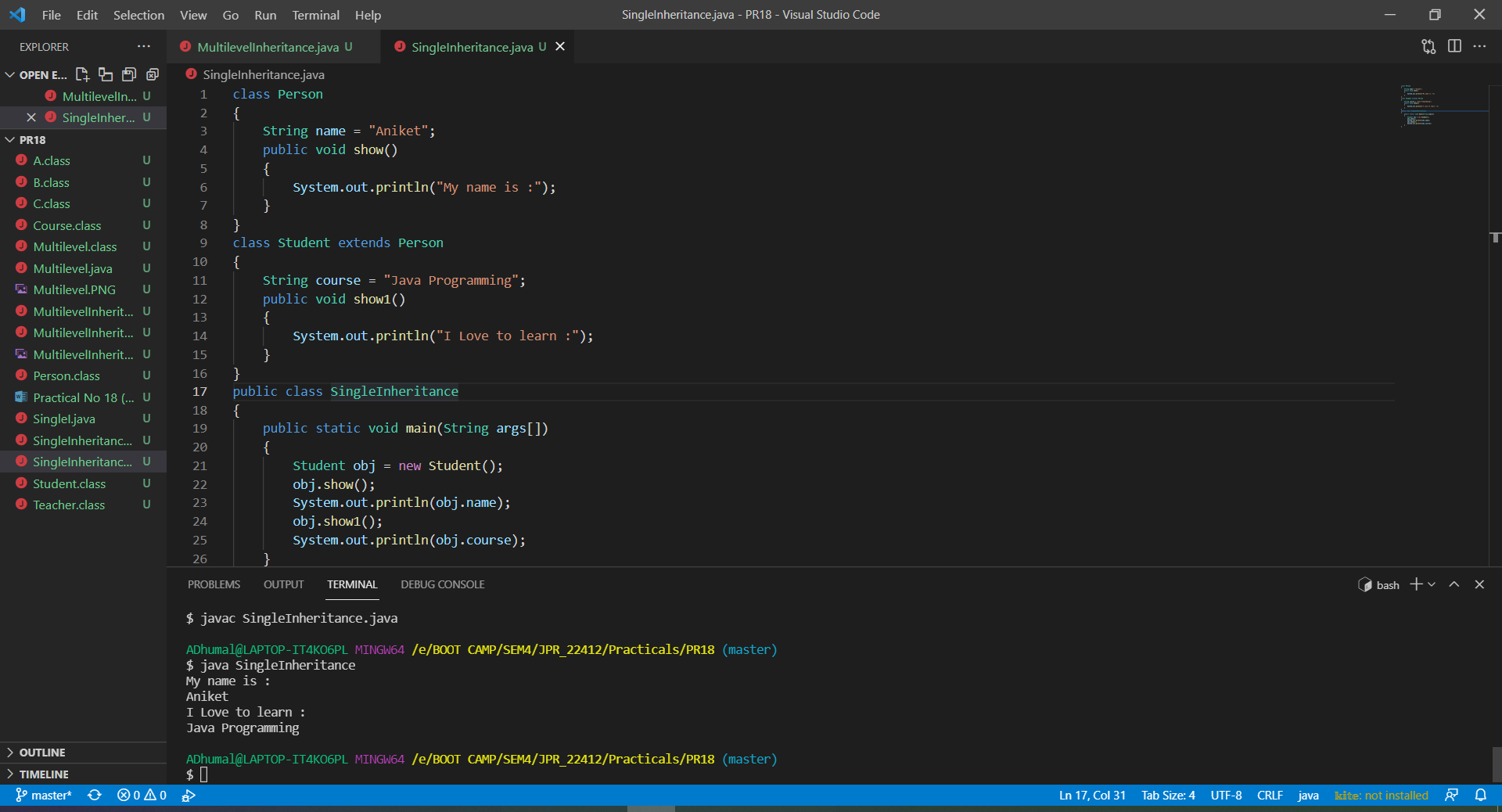
c.disC();

}

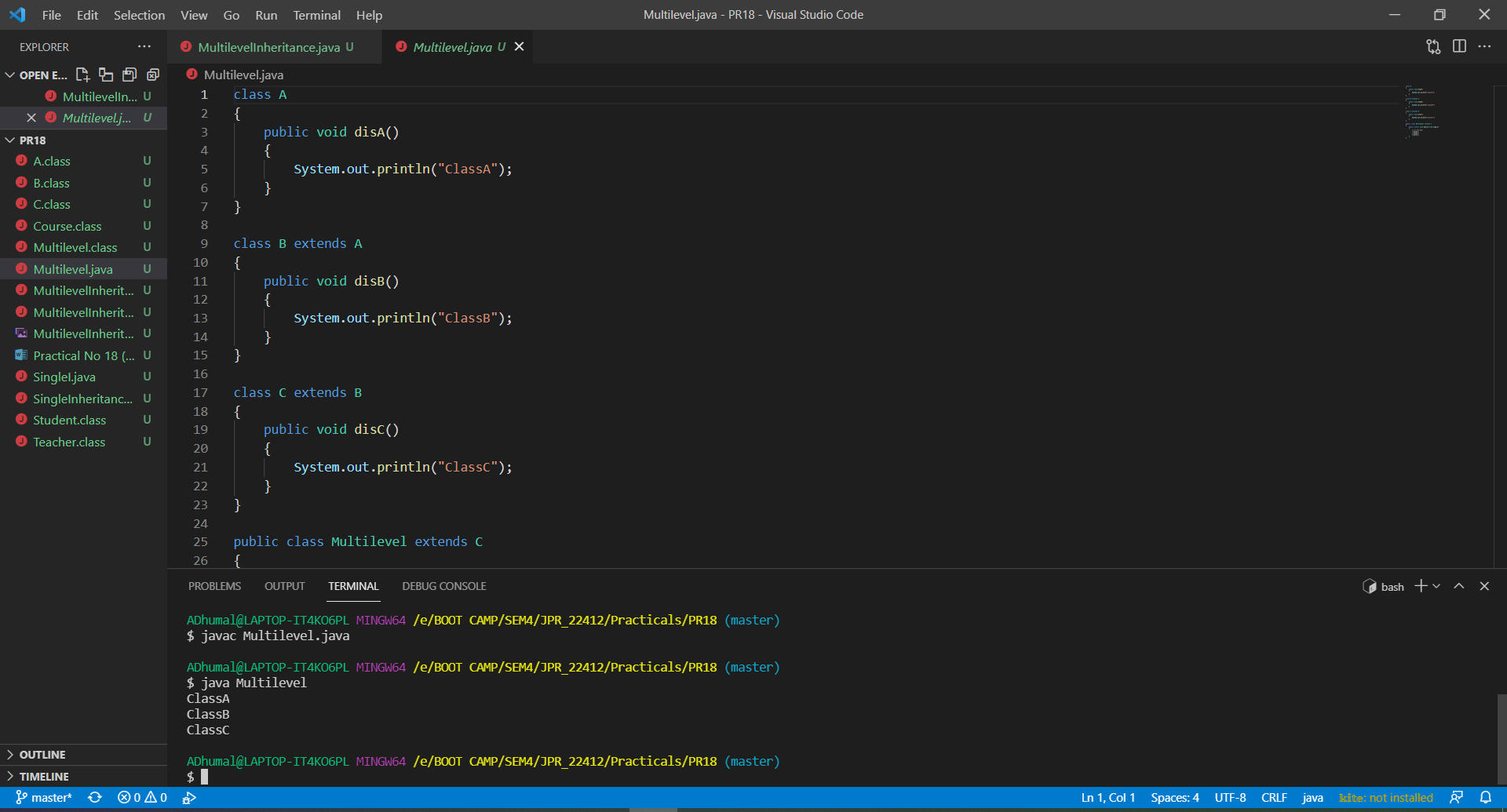
}

1. **Result:**

**Single Inheritance**



**Multilevel Inheritance**

****

1. **Practical Related Questions:**
2. **Justify: Java does not support multiple inheritance.**

* In java, multiple inheritance is not supported because of ambiguity problem.
* Consider a case where class Multiple extends class1 and class2 and both have the same method display().
* Now java computer cannot decide, which display method it should inherit. To prevent such situation, multiple inheritances is not allowed in java.

1. **Specify the conditions when the super keyword can be used.**

Super can be used to refer immediate parent class instance variable. super can be used to invoked immediate parent class method. super() can be used to invoke immediate parent class constructor.

1. **Write the importance of final variables and methods.**

When an anonymous inner class is defined within the body of a method, all variables declared final in the scope of that method are accessible from within the inner class. For scalar values, once it has been assigned, the value of the final variable cannot change. For object values, the reference cannot change.

1. **Specify the conditions which needs to be satisfied while using the abstract classes.**

Abstract method has no body. Always end the declaration with a semicolon (;). It must be overridden. An abstract class must be extended and in a same way abstract method must be overridden.

1. **Exercise:**
2. **Develop program to implement the multilevel inheritance.**

class Student

{

public void Info1()

{

System.out.println("JAVA Students.");

}

}

class Course extends Student

{

public void Info2()

{

System.out.println("Students are learning java programming");

}

}

class Teacher extends Course

{

public void Info3()

{

System.out.println("Teacher Teaching java.");

}

}

class MultilevelInheritance

{

public static void main(String args[])

{

Teacher t = new Teacher();

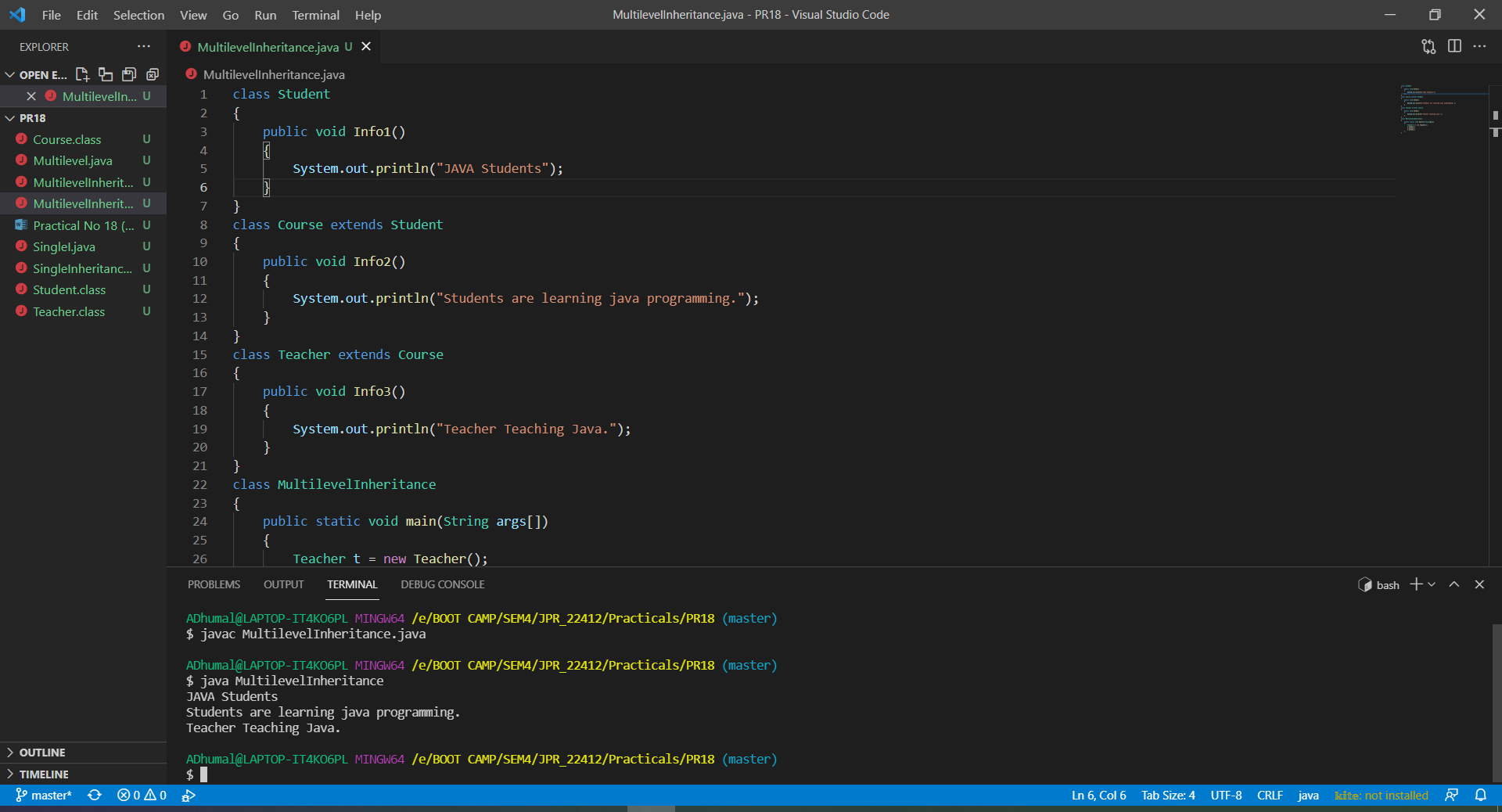
t.Info1();

t.Info2();

t.Info3();

}

}



1. **Develop a program to calculate the room area and volume to illustrate the concept of single inheritance.**

class Room

{

int length;

int breadth;

Room(int x,int y)

{

length=x;

breadth=y;

}

int area()

{

return(length\*breadth);

}

}

class BedRoom extends Room

{

int height;

BedRoom(int x,int y,int z)

{

super(x,y);

height=z;

}

int volume()

{

return (length\*breadth\*height);

}

}

class SingleI

{

public static void main(String args[])

{

BedRoom room=new BedRoom(14,12,10);

int area=room.area();

int volume=room.volume();

System.out.println(area);

System.out.println(volume);

}

}

