Lab – 5

By:- Manav Dhiman

Roll No. – 19103063

Q1: Write a java program to demonstrate the difference amongst constructor,  static and non static initializer blocks. Your program should have at least one constructor in super and subclass

class A{

    A(){

        System.out.println("Hello from  A");

    }

    {System.out.println("I'm non-static Block-A");}

    static {

        System.out.println("I'm static block-A");

    }

};

class B extends A{

    B(){

        System.out.println("Hello from B");

    }

    {System.out.println("I'm non-static Block-B");}

    static {

        System.out.println("I'm static block-B");

    }

}

class C{

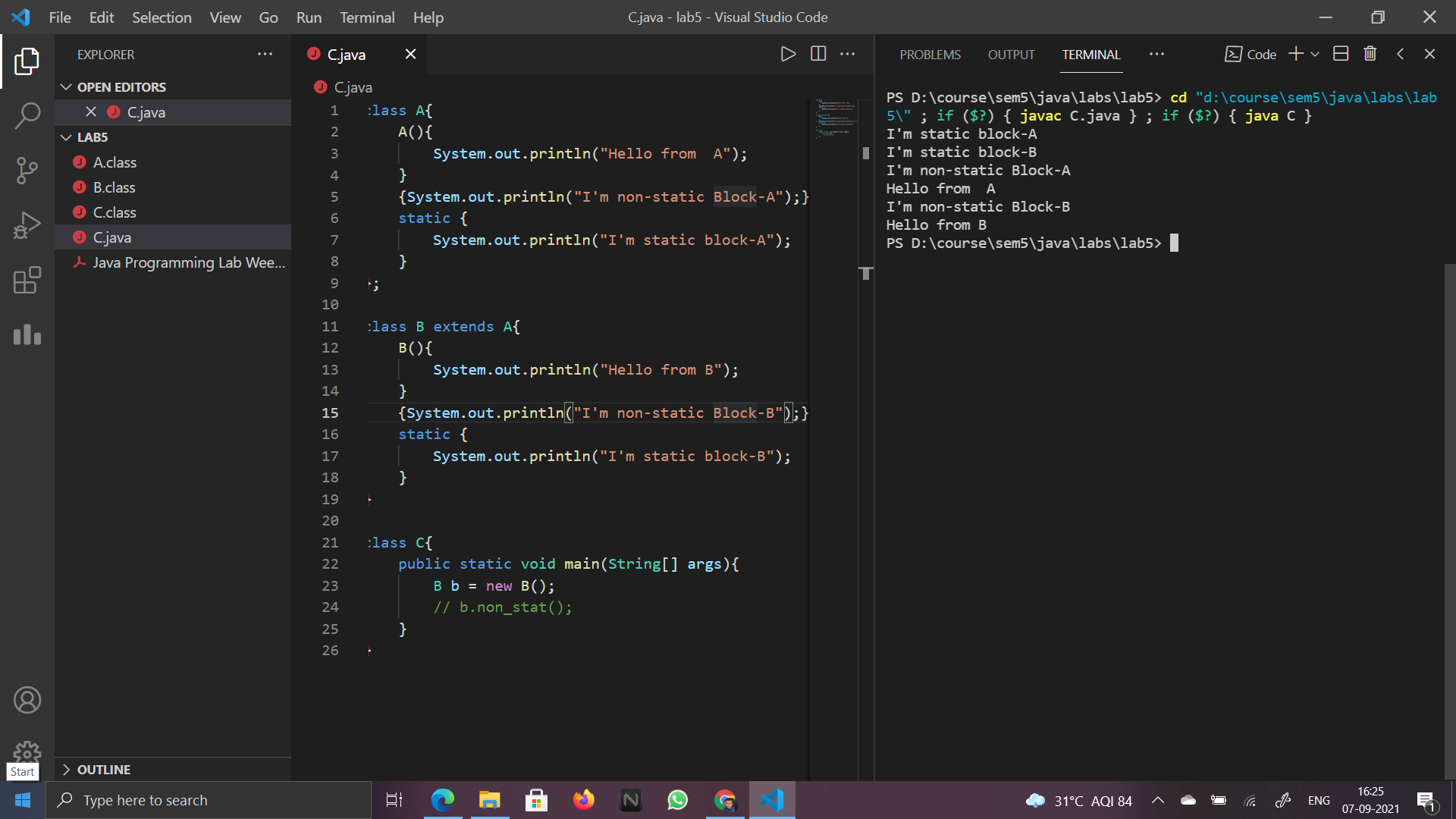
    public static void main(String[] args){

        B b = new B();

        // b.non\_stat();

    }

}



Q2. Write a program to show the usefulness of abstract classes. Inherit an abstract class in another abstract class and implement the methods of the child abstract class in another class and show the calling of the methods defined in abstract classes using dynamic method dispatching. Your abstract classes must have constructors.

abstract class A{

    String s;

    A(String c){

        s=c;

        System.out.println("Constructor of A");

    }

    abstract void f();

};

abstract class B extends A{

    B(String c){

        super(c);

        s=c;

        System.out.println("Constructor of B");

    }

    abstract void f();

};

class C extends B{

    C(String s1){

        super(s1);

        s=s1;

        System.out.println("Constructor of C");

    }

    void f(){

        System.out.println("String is-> "+s);

    }

};

class DynamicDisp{

    public static void main(String args[]){

        C c2 = new C("Check for B");

        B b1 = c2;

        b1.f();

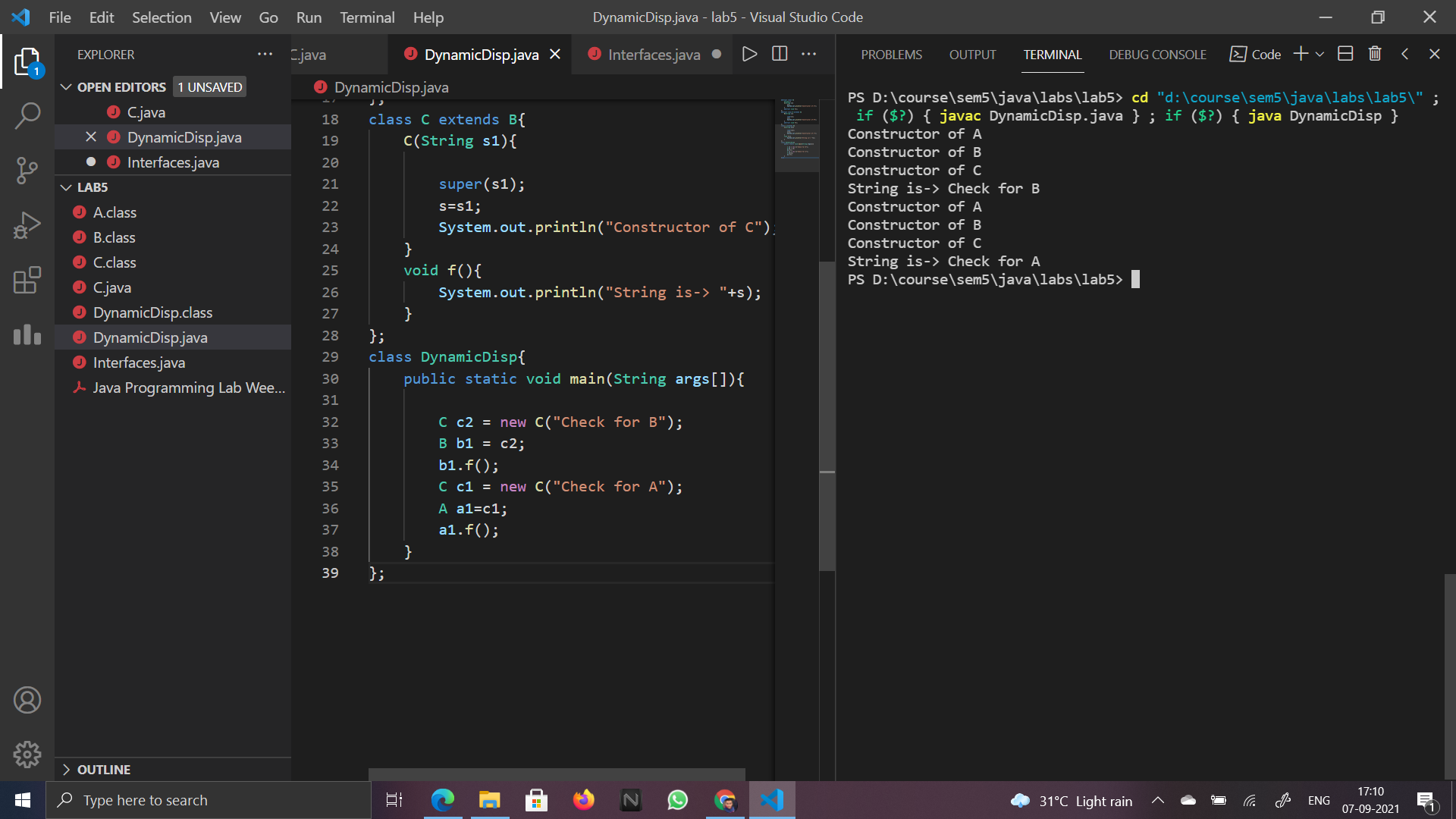
        C c1 = new C("Check for A");

        A a1=c1;

        a1.f();

    }

};



Q3. Write a program to demonstrate the use of interfaces. Add the methods and fields in an interface and implement that interface in a class

interface A{

    void f();

}

class B implements A{

    public void f(){

        System.out.println("Hello from B");

    }

};

class Interfaces{

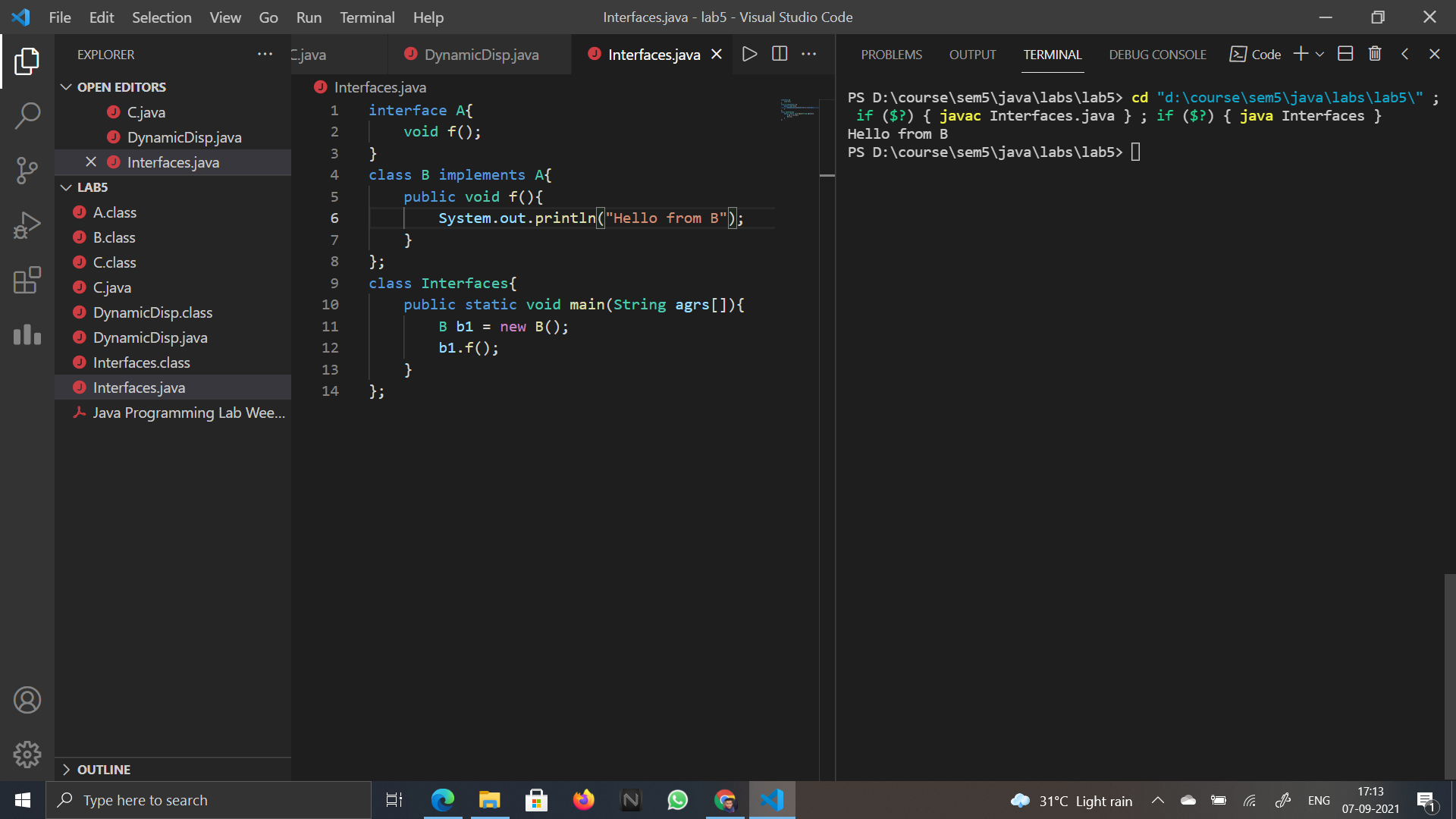
    public static void main(String agrs[]){

        B b1 = new B();

        b1.f();

    }

};



Q4. How do you achieve multiple inheritance in java using interfaces? Write a program to show the multiple inheritance implementation using interfaces.

interface A{

    void f();

}

interface B{

    void fun();

}

class C implements A,B{

    public void f(){

        System.out.println("Hello from f");

    }

    public void fun(){

        System.out.println("Hello from fun");

    }

};

class Multiple{

    public static void main(String agrs[]){

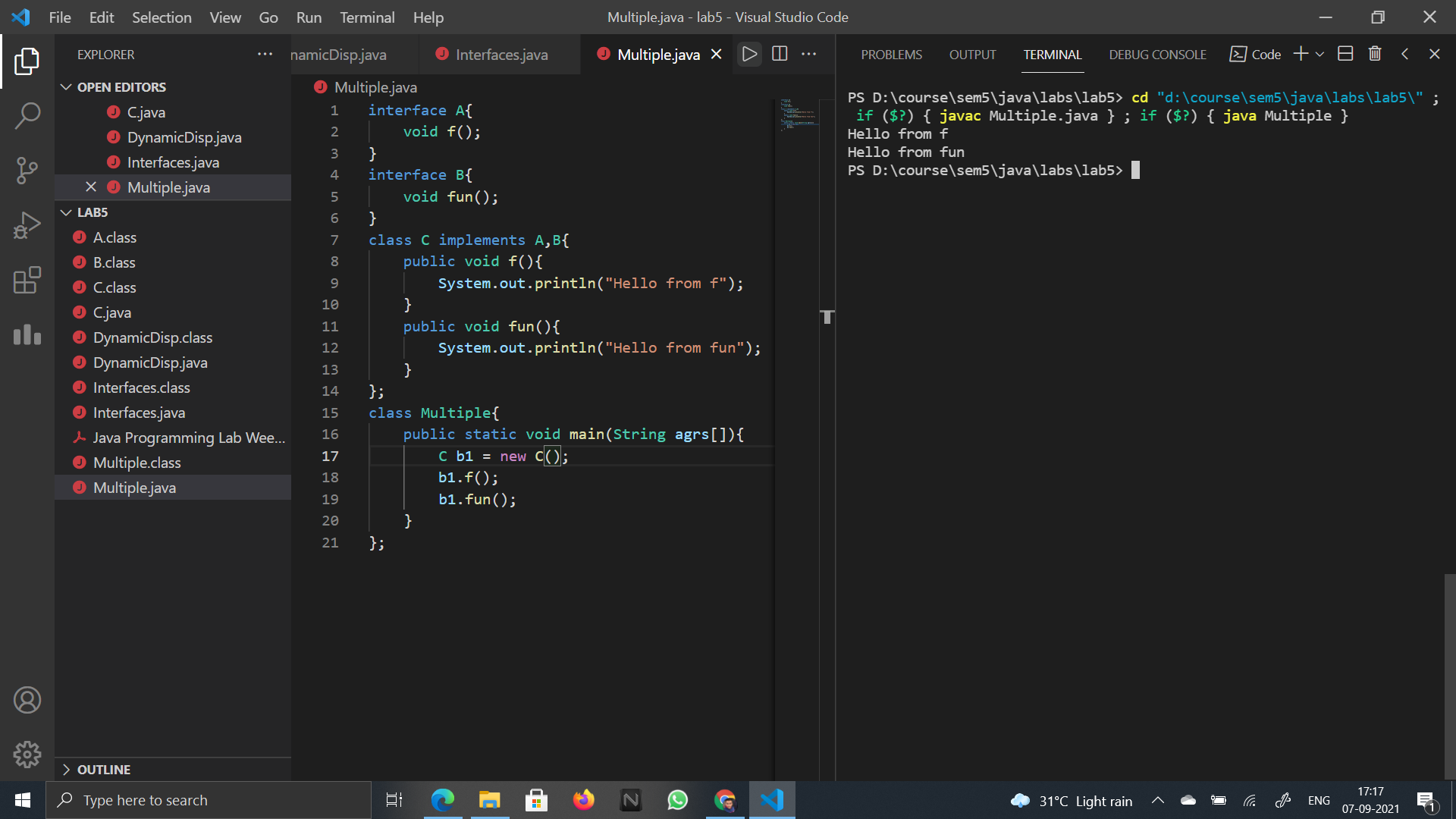
        C b1 = new C();

        b1.f();

        b1.fun();

    }

};



Q5 Demonstrate how an abstract class can inherit an interface and another abstract class

interface A{

    void f();

}

abstract class B{

    abstract void fun();

};

abstract class C extends B implements A{

    public void f(){

        System.out.println("From abstract C");

    }

    abstract void fun();

};

class D extends C{

    public void f(){

        System.out.println("I'm from extends");

    }

    public void fun(){

        System.out.println("I'm from abstract B");

    }

};

class InheritInterface{

    public static void main(String agrs[]){

        D b1 = new D();

        b1.f();

        b1.fun();

    }

};

