SCHOOL OF COMPUTING

M.REDDY MANISH

CH.SC.U4CSE24129

OBJECT ORIENTED PROGRAMMING (23CSE111)

LAB RECORD

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, CHENNAI

BONAFIDE CERTIFICATE

This is to certify that the Lab Record work for 23CSE111- Object Oriented Programming Subject submitted by ***CH.SC.U4CSE24129 – M. Reddy Manish*** in **"Computer Science and Engineering"** is a Bonafide record of the work carried out under my guidance and supervision at Amrita School of Computing, Chennai.

This Lab examination held on

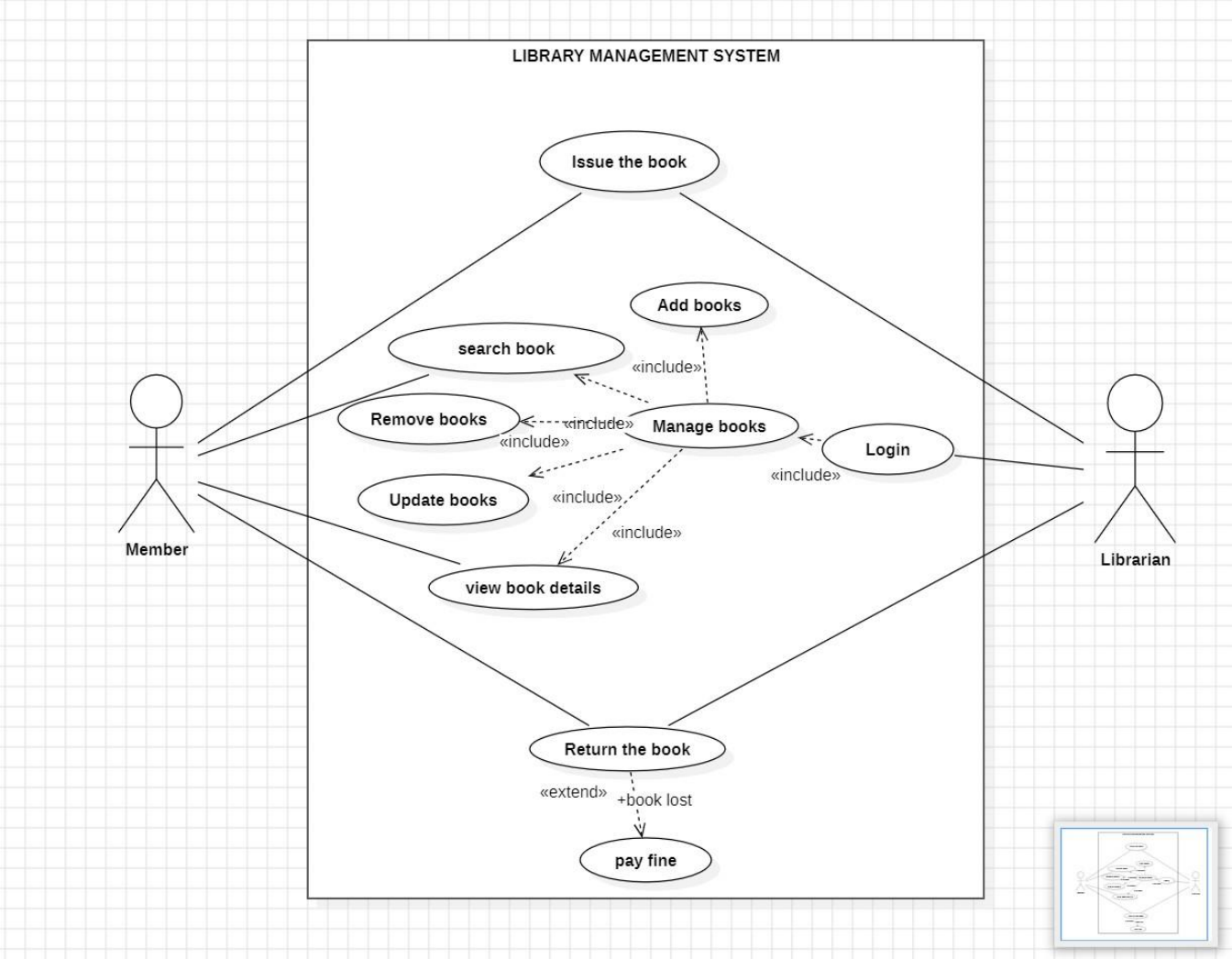
Internal Examiner 1 Internal Examiner 2

INDEX

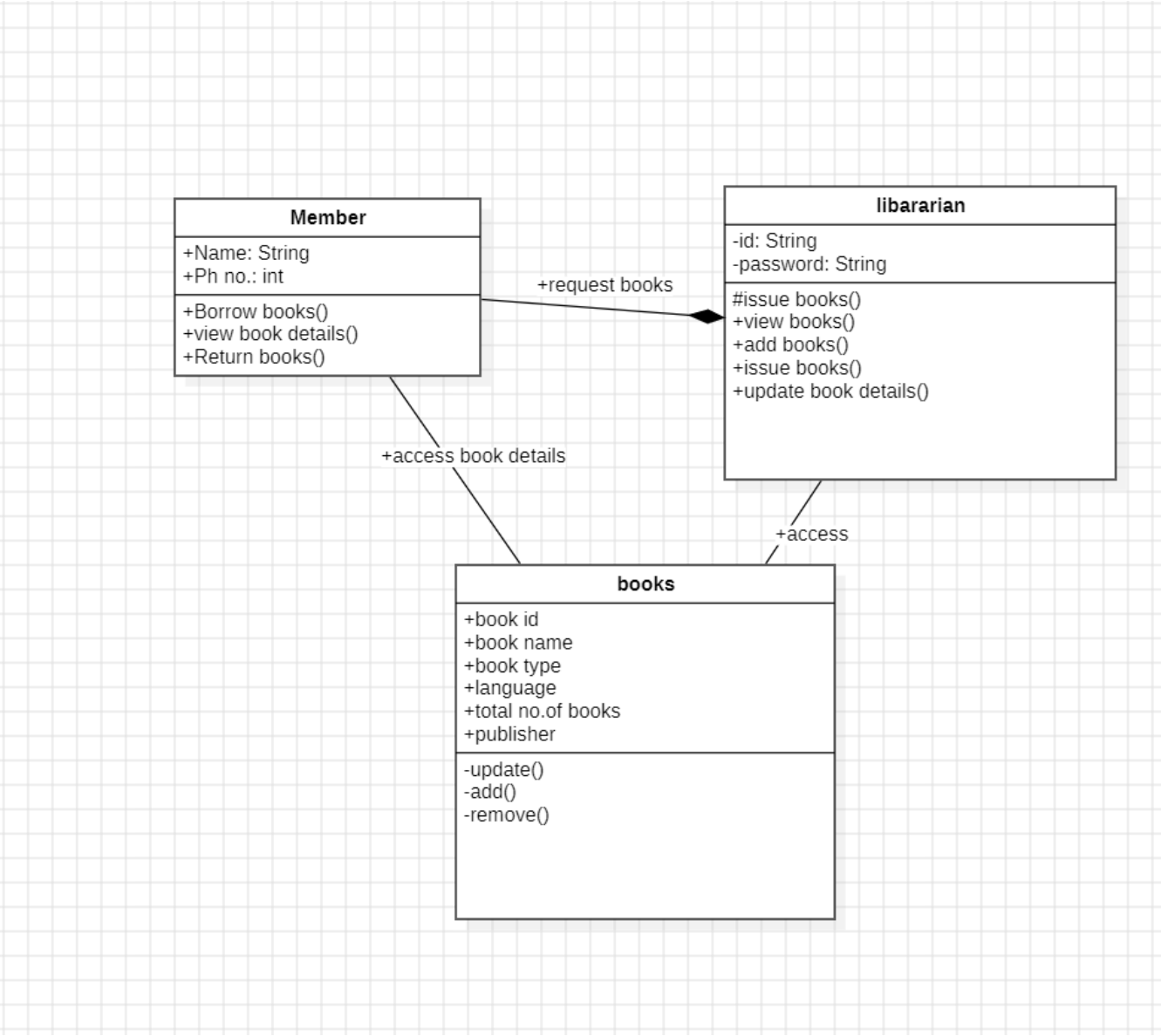
UML DIAGRAMS

1. Library Management system

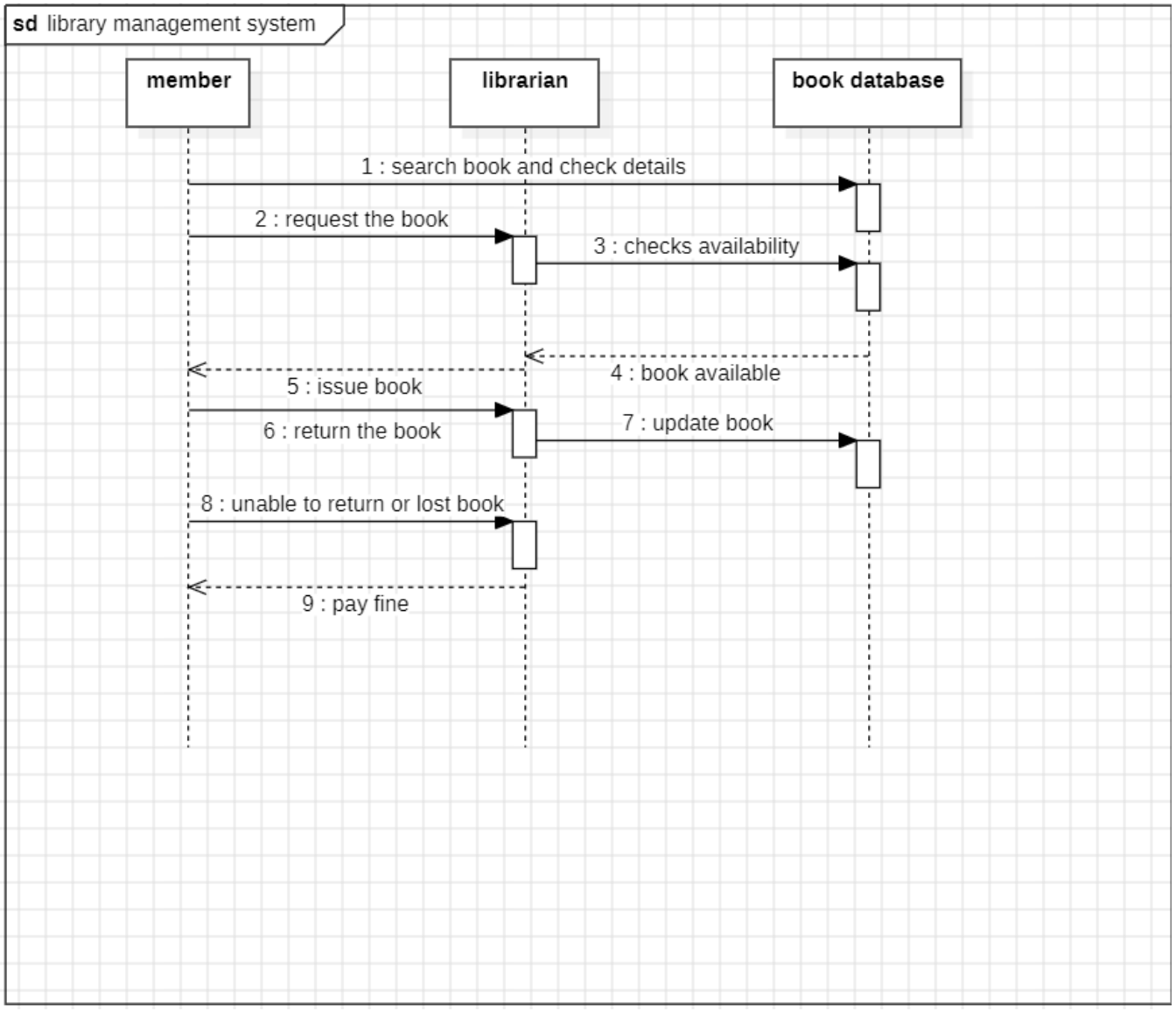
1.a) Use Case Diagram



1.b) Class Diagram

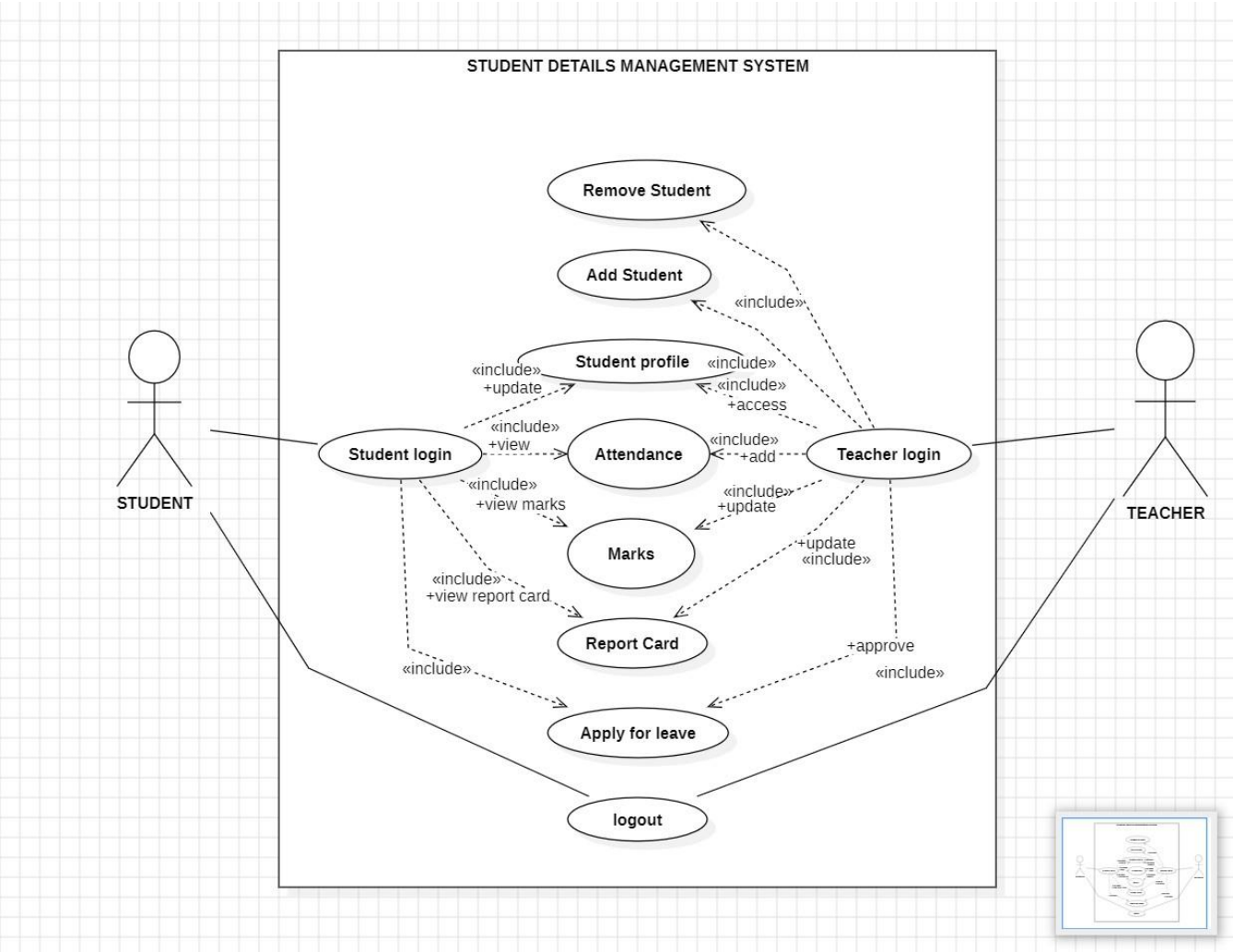


1.c) Sequence Diagram

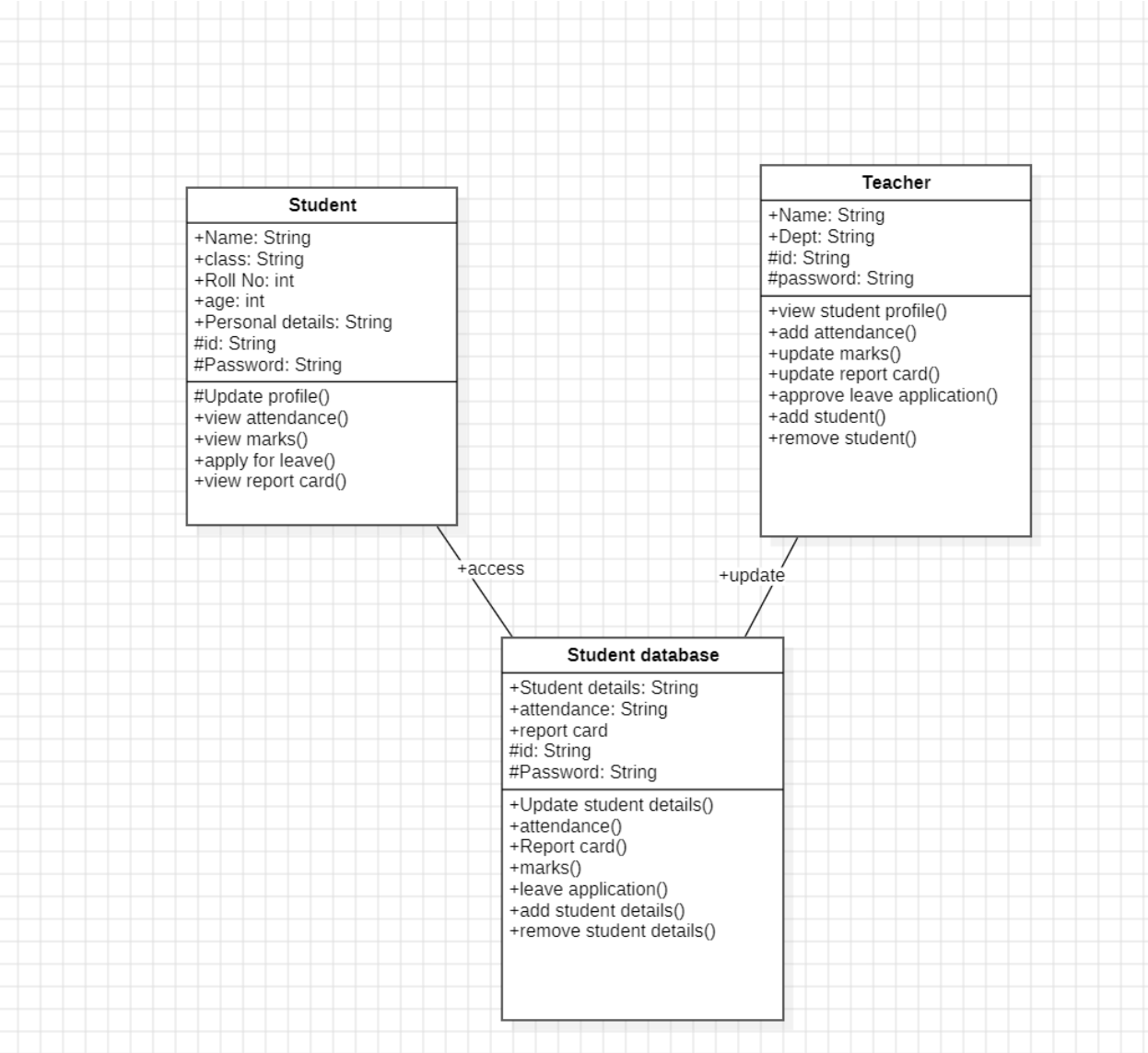


2. Student Management system

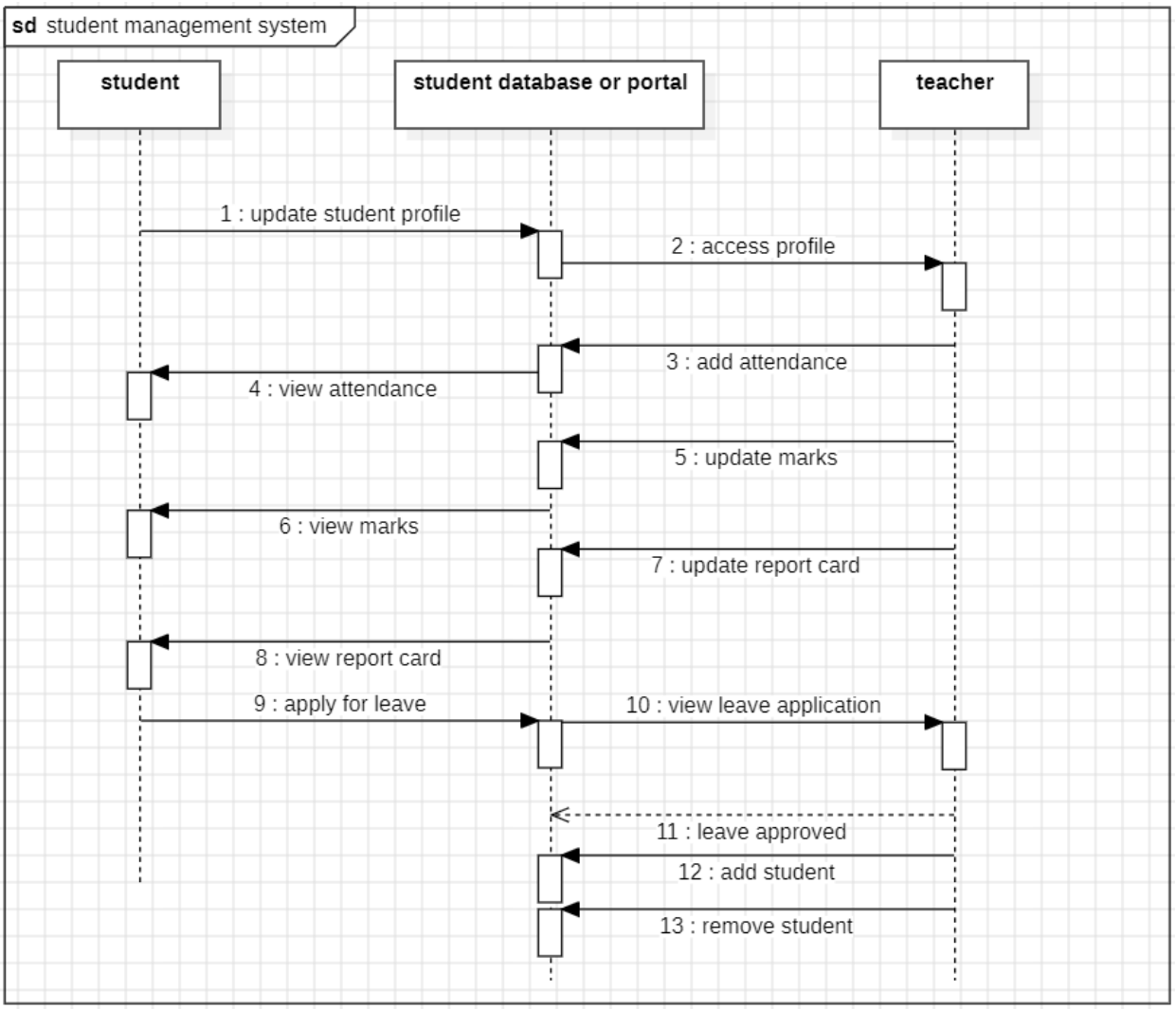
2.a) Use Case Diagram



2.b) Class Diagram

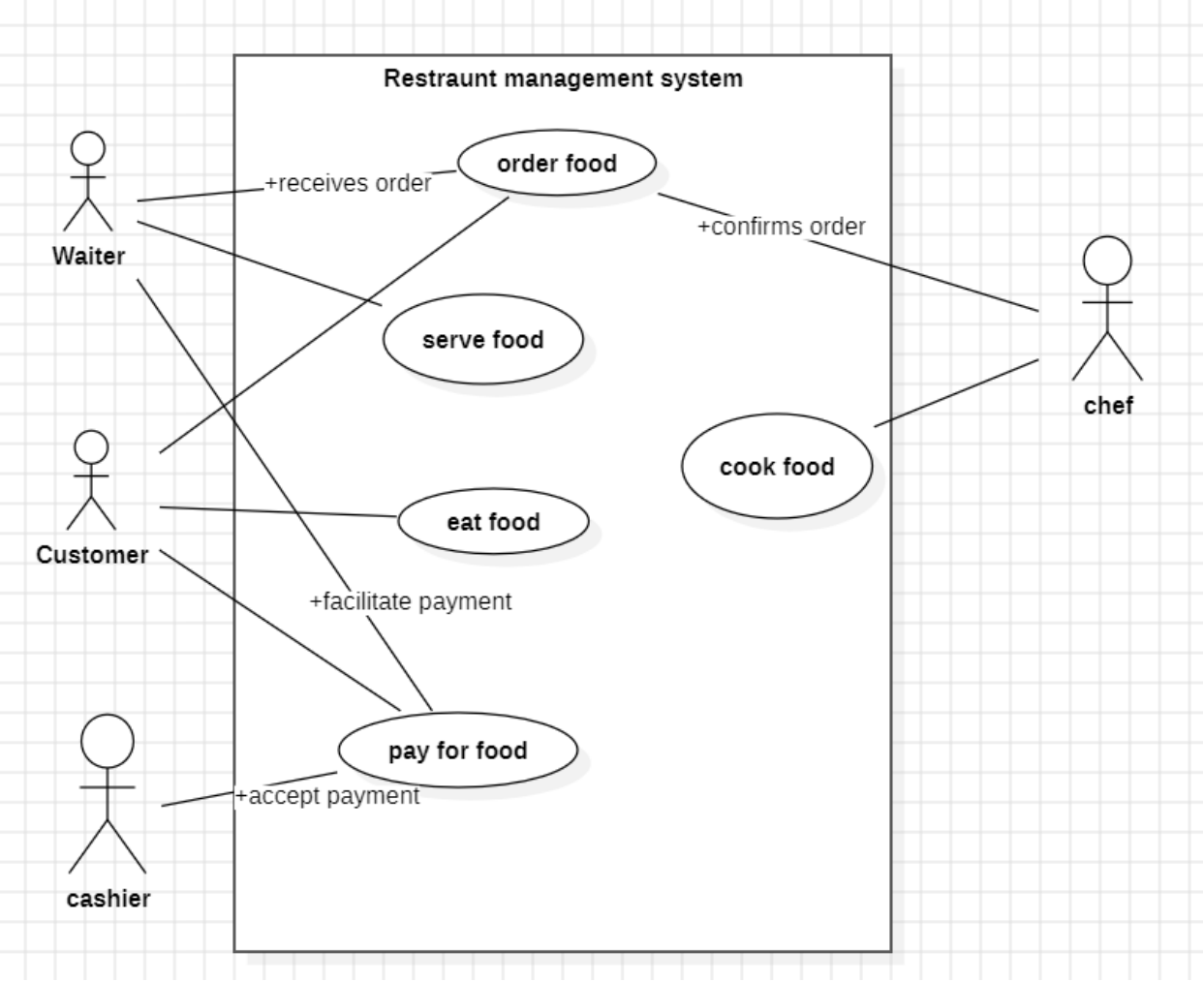


2.c) Sequence Diagram

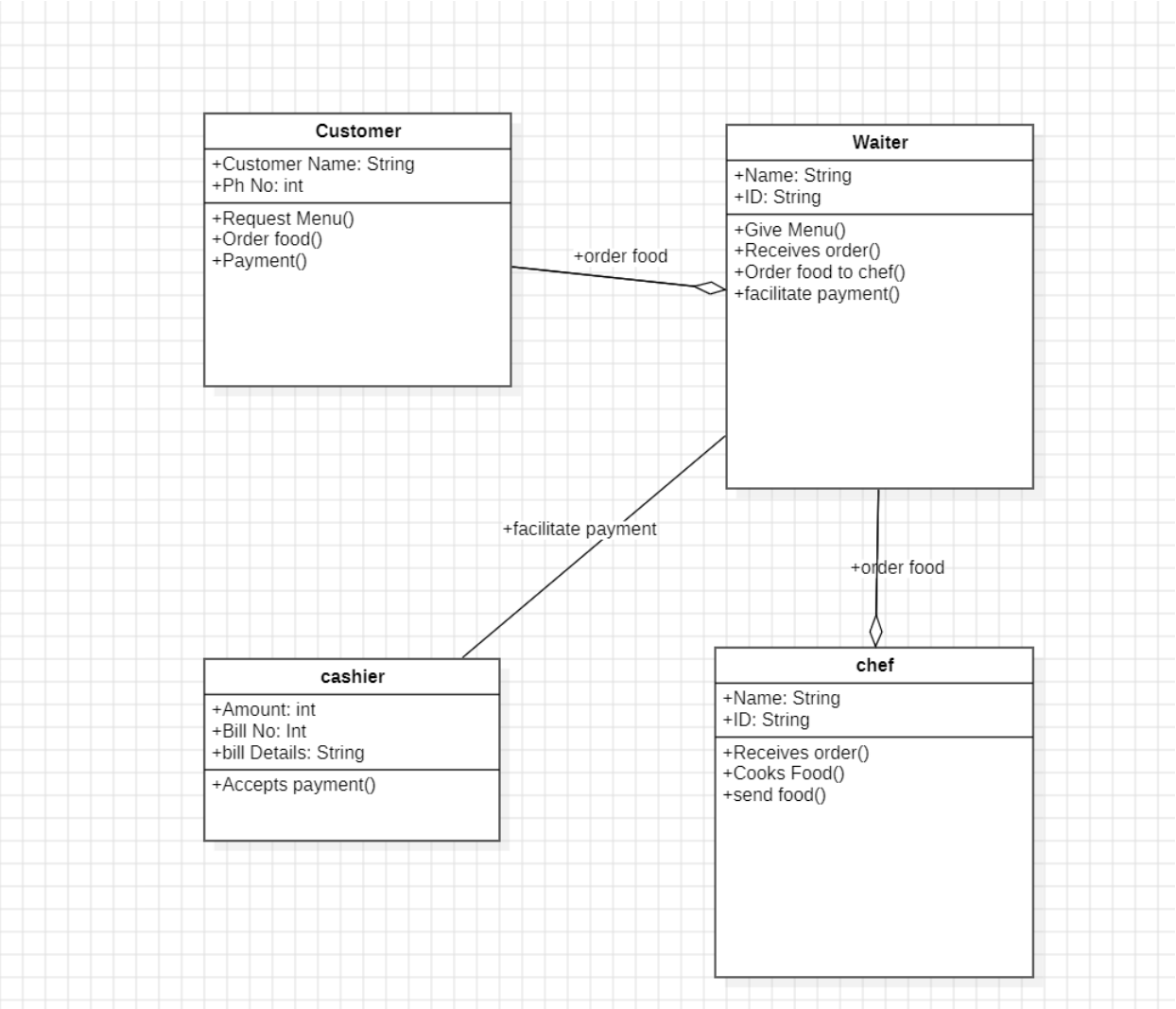


3. Restaurent Management system

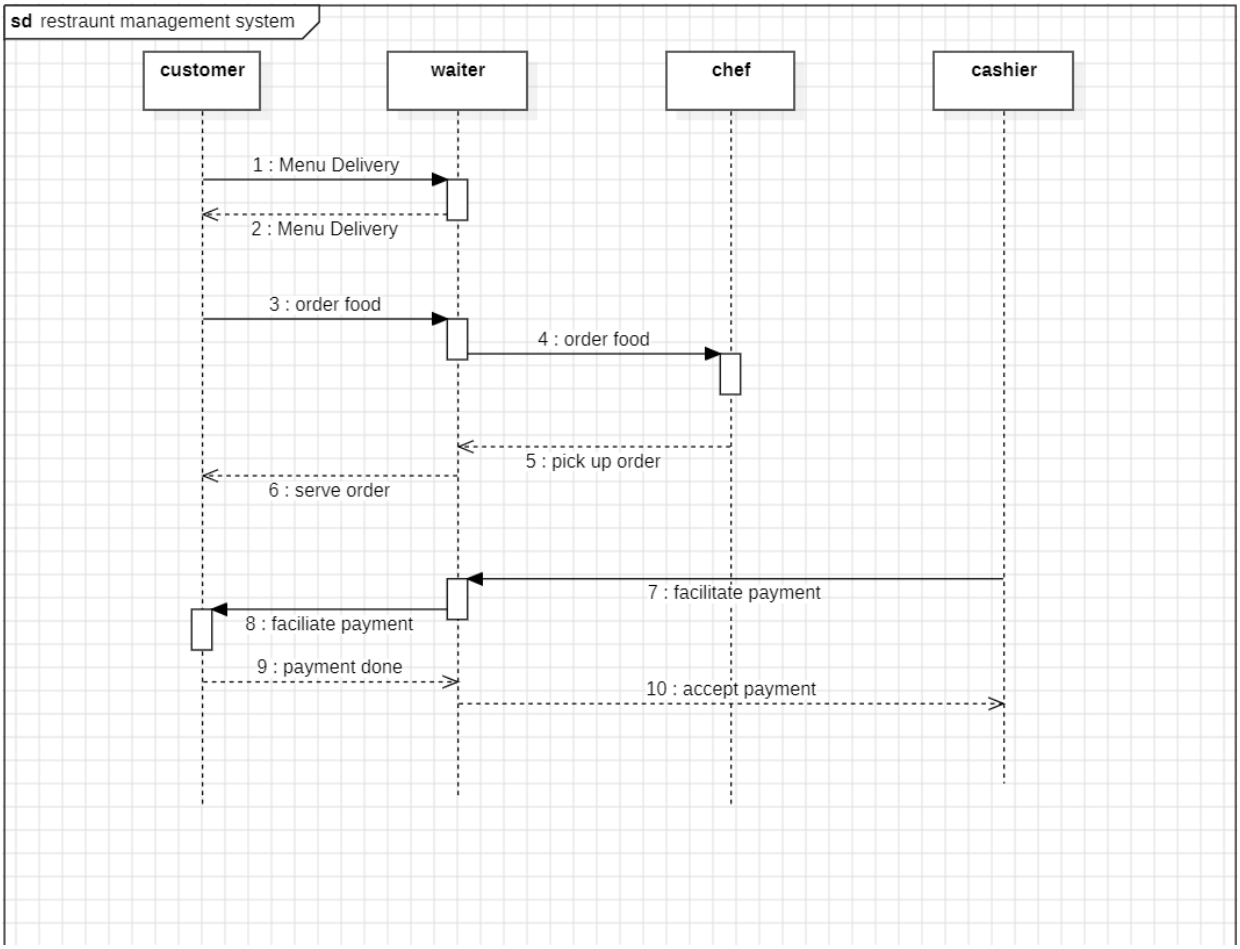
3.a) Use Case Diagram



3.b) Class Diagram



3.c) Sequence Diagram



BASIC JAVA PROGRAMS

4.a. JAVA PROGRAME TO FIND HYPOTENUSE OF A TRIANGLE:-

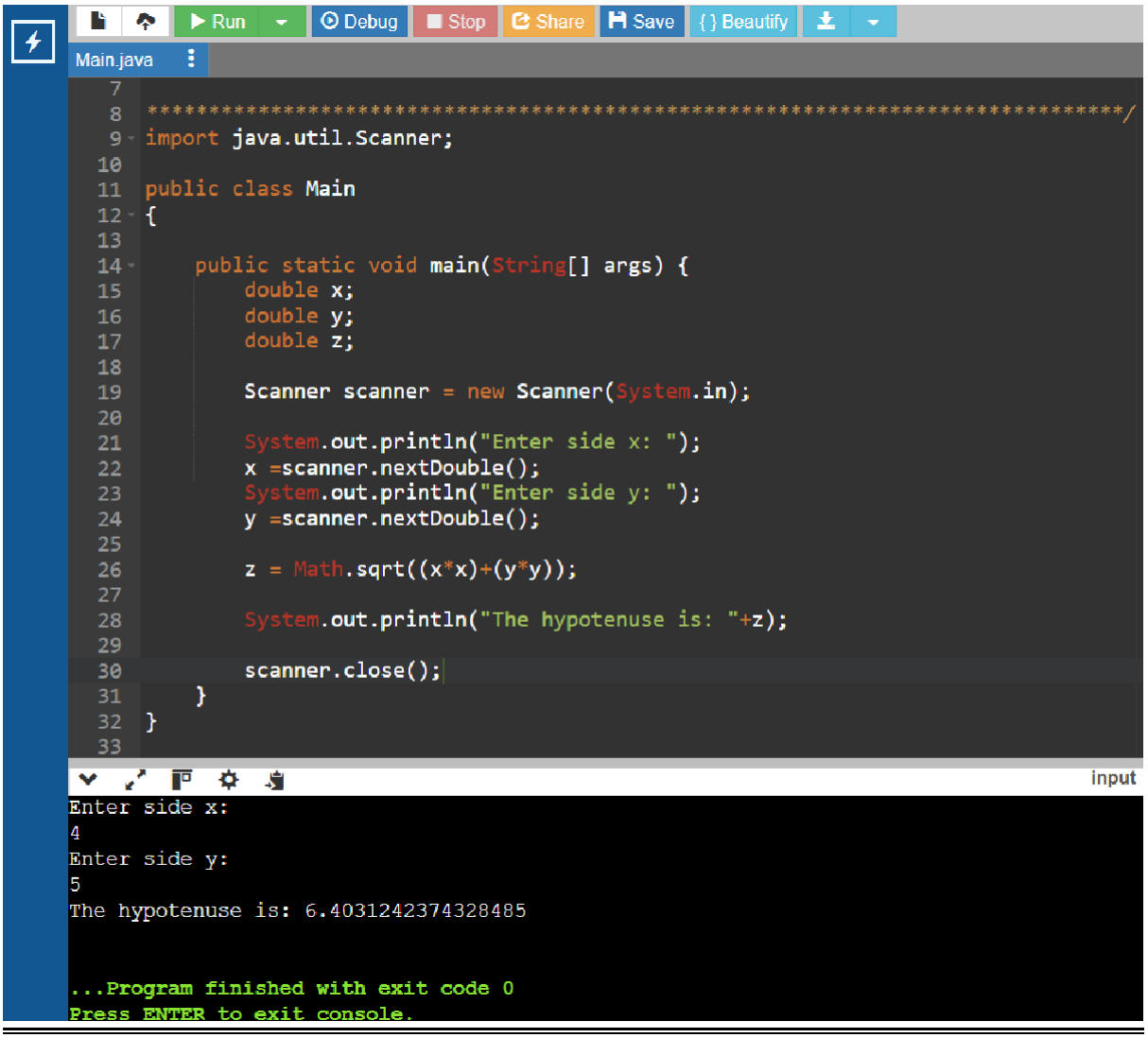
AIM:- To Find Hypotenuse of a Triangle

JAVA CODE:-

```
import java.util.Scanner;


public class Main
{

    public static void main(String[] args) {
        double x;
        double y;
        double z;

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter side x: ");
        x =scanner.nextDouble();
      System.out.println("Enter side y: ");
      y =scanner.nextDouble();

       z = Math.sqrt((x*x)+(y*y));
     System.out.println("The hypotenuse is: "+z);
    scanner.close();
    }
```

}

SCREENSHOT WITH OUTPUT



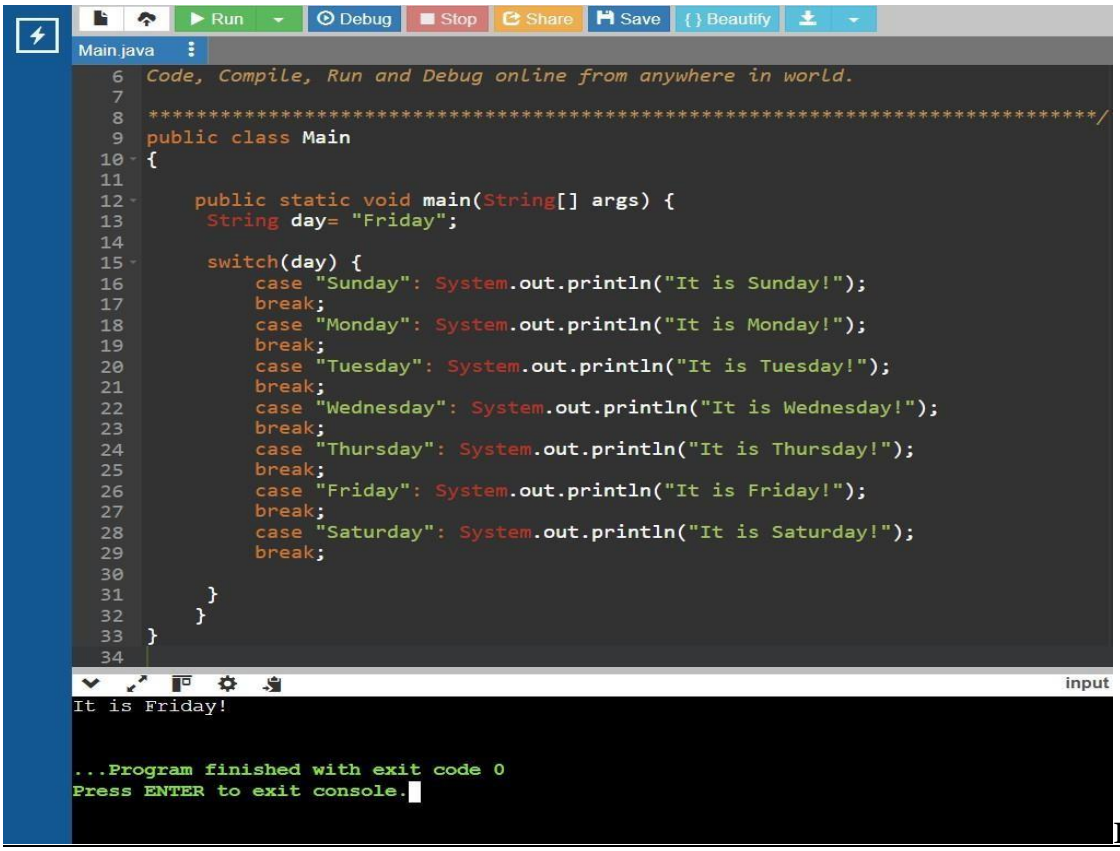
4.b. JAVA PROGRAME WITH SWITCH CASE:-

AIM:- To Explore the switch case Element In the java programe

JAVA CODE:-

```
public class Main
{
   public static void main(String[] args) {
            String day= "Friday";
             switch(day) {
               case "Sunday": System.out.println("It is Sunday!");
               break;
               case "Monday": System.out.println("It is Monday!");
               break;
               case "Tuesday": System.out.println("It is Tuesday!");
               break;
               case "Wednesday": System.out.println("It is Wednesday!");
               break;
               case "Thursday": System.out.println("It is Thursday!");
               break;
               case "Friday": System.out.println("It is Friday!");
               break;
               case "Saturday": System.out.println("It is Saturday!");
               break;

           }
          }
}
```

SCREENSHOT WITH OUTPUT



PU

4.c. JAVA PROGRAME USING IF ELSE STATEMENTS:-

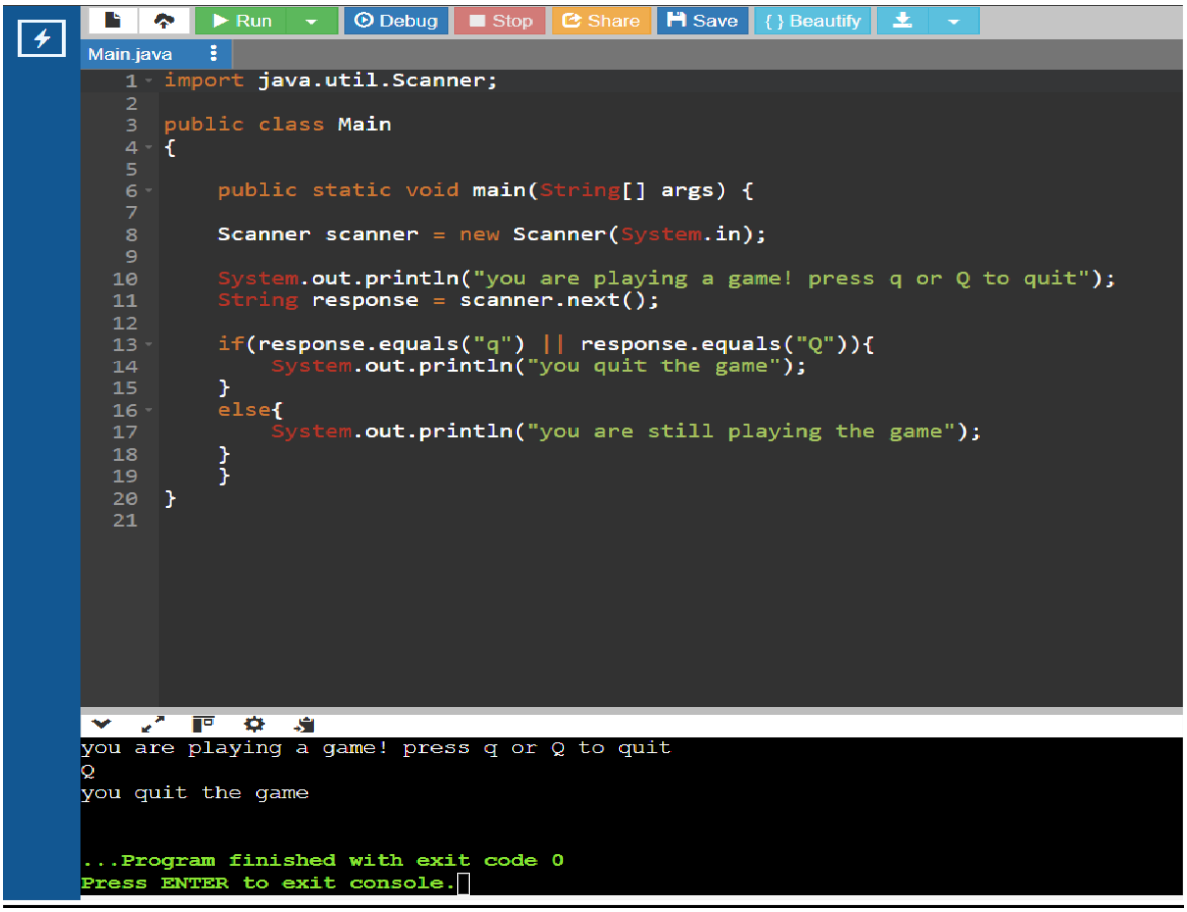
AIM:- To Decode the If else statements in java code

JAVA CODE:-

```
import java.util.Scanner;


public class Main
{

        public static void main(String[] args) {


        Scanner scanner = new Scanner(System.in);
```

```
        System.out.println("you are playing a game! press q or Q to quit");
        String response = scanner.next();

        if(response.equals("q") || response.equals("Q")){
           System.out.println("you quit the game");
        }
        else{
           System.out.println("you are still playing the game");
        }
        }
}
```

SCREENSHOT WITH OUTPUT:-



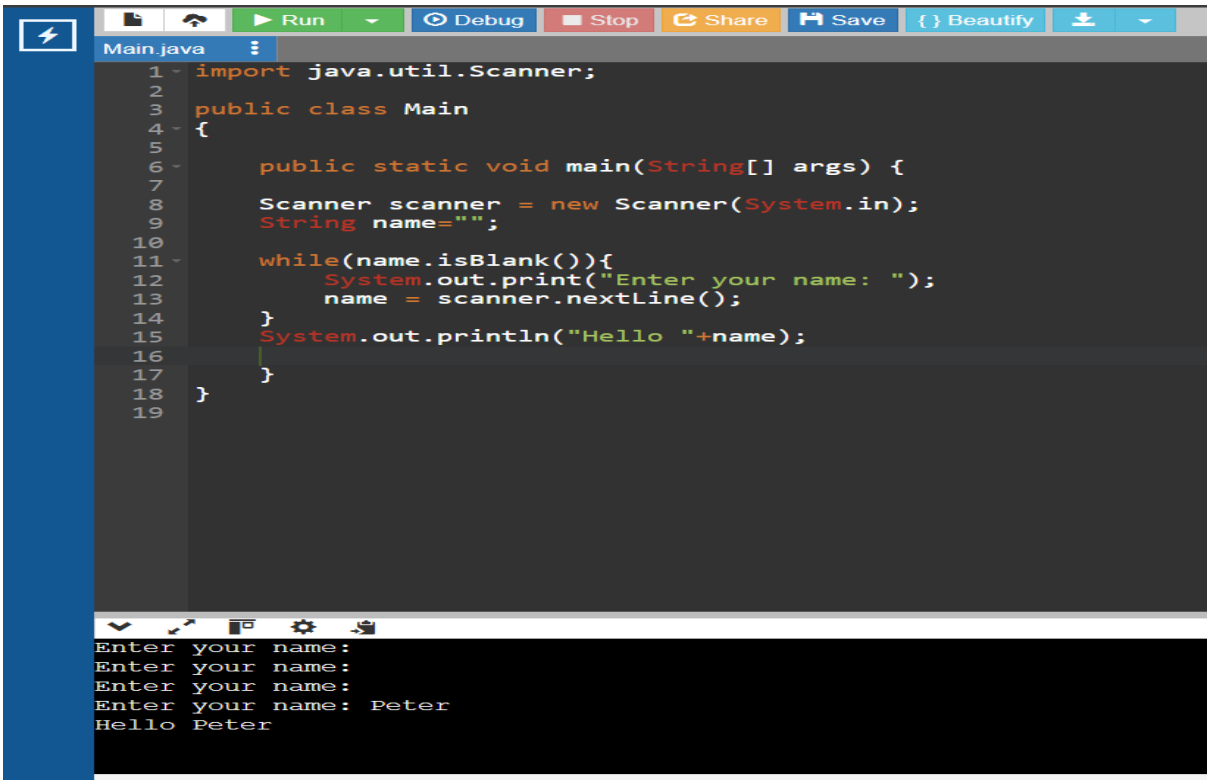
4.D.JAVA PROGRAME USING WHILE LOOP:-

AIM:- To Decode the while loops in java code

JAVA CODE:-

```
import java.util.Scanner;


public class Main
{

	public static void main(String[] args) {


	Scanner scanner = new Scanner(System.in);
	String name="";


	while(name.isBlank()){
	   System.out.print("Enter your name: ");
	   name = scanner.nextLine();
	}
	System.out.println("Hello "+name);


	}
}
```

SCREENSHOT WITH OUTPUT:-



4.e. JAVA PROGRAME USING FOR LOOP:-

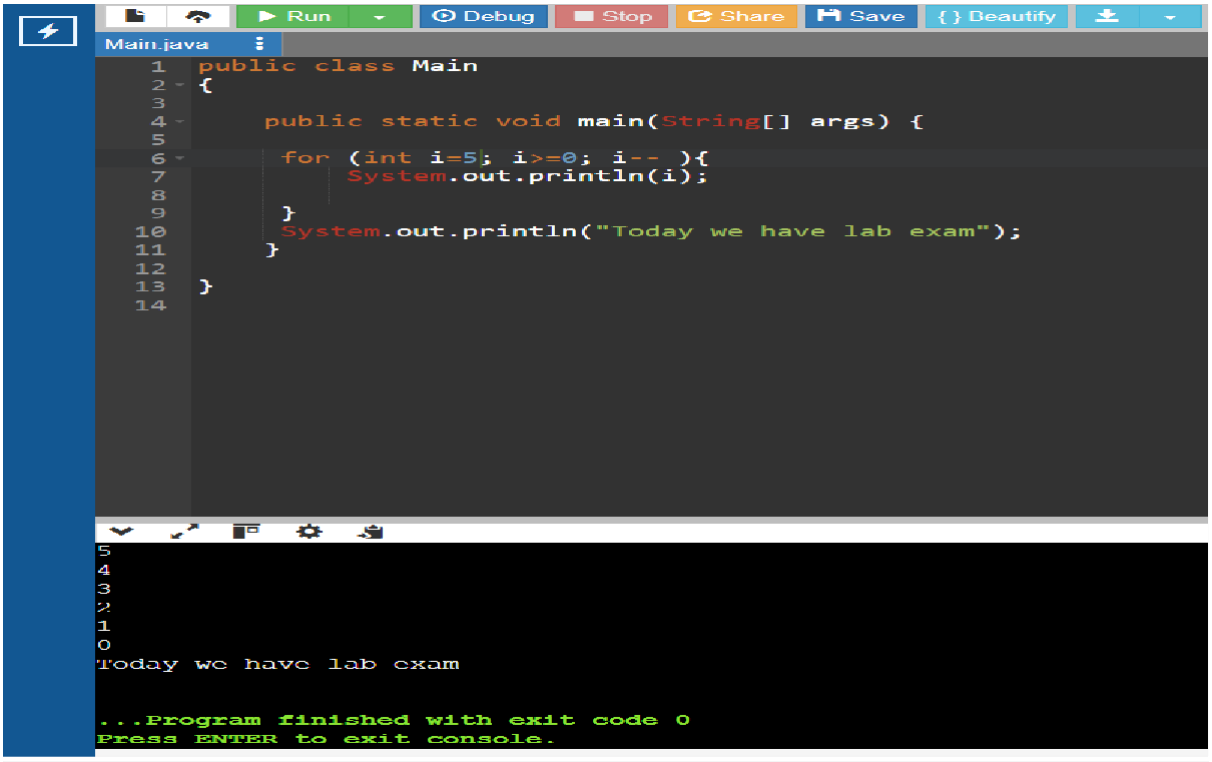
AIM:- To Decode the For loops in java code

JAVA CODE:-

```
public class Main
{

        public static void main(String[] args) {


   for (int i=5; i>=0; i-- ){
      System.out.println(i);


   }
```

```
    System.out.println("Today we have lab exam");
        }
}
```

SCREENSHOT WITH OUTPUT:-



4.f. JAVA PROGRAME FOR SHAPE OF A SYMBOL:-

AIM:- To Find the shape of any symbol using java programe.

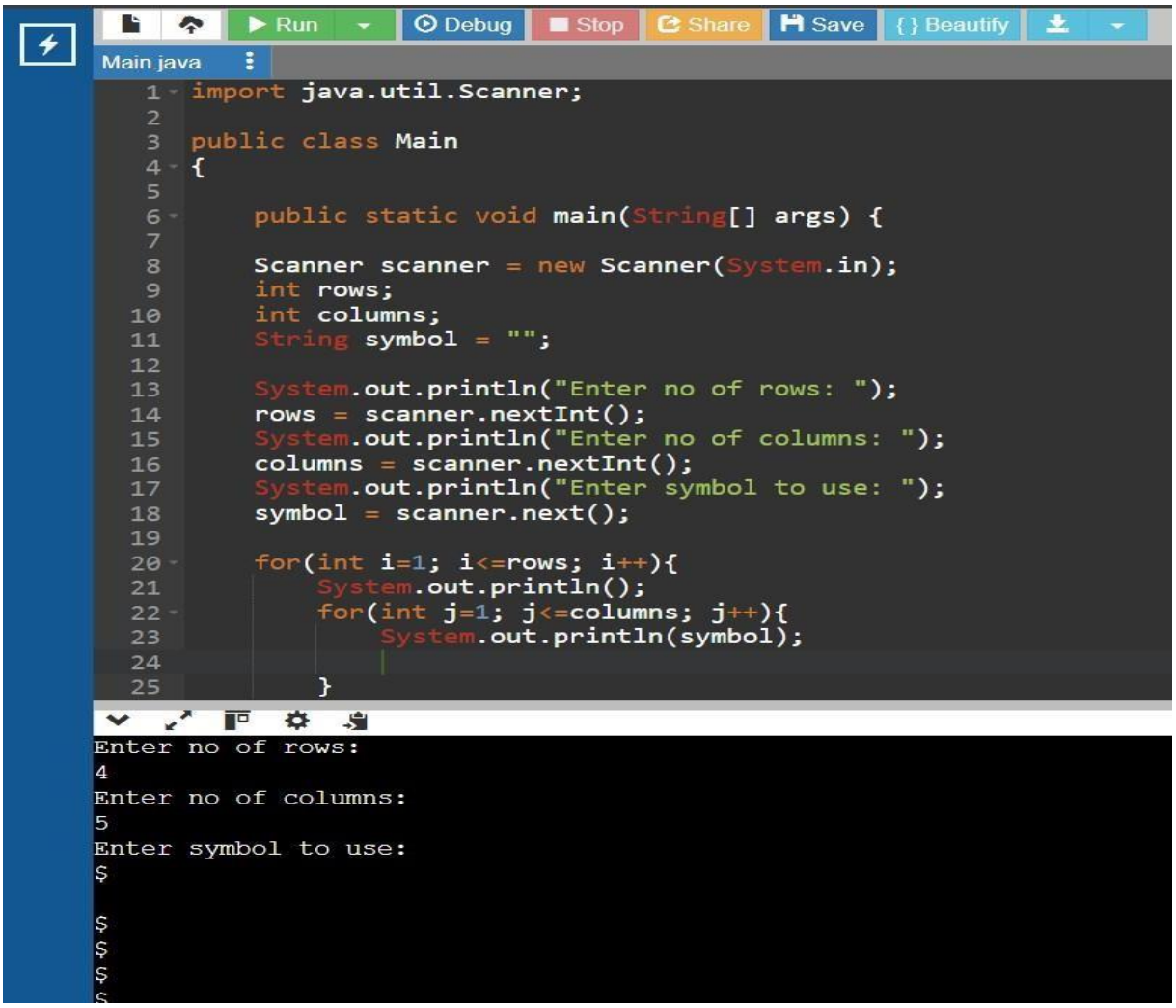
JAVA CODE:-

```
import java.util.Scanner;


public class Main

{


        public static void main(String[] args) {
```

```
Scanner scanner = new Scanner(System.in);
int rows;
int columns;
```

```
    String symbol = "";

    System.out.println("Enter no of rows: ");
    rows = scanner.nextInt();
    System.out.println("Enter no of columns: ");
    columns = scanner.nextInt();
    System.out.println("Enter symbol to use: ");
    symbol = scanner.next();

    for(int i=1; i<=rows; i++){
        System.out.println();
        for(int j=1; j<=columns; j++){
            System.out.println(symbol);

        }
    }
            }
}
```

SCREENSHOT WITH OUTPUT:-



4.g. JAVA PROGRAME FOR FACTORIAL OF A NUMBER:-

AIM:- To Find the Factorial of a Number

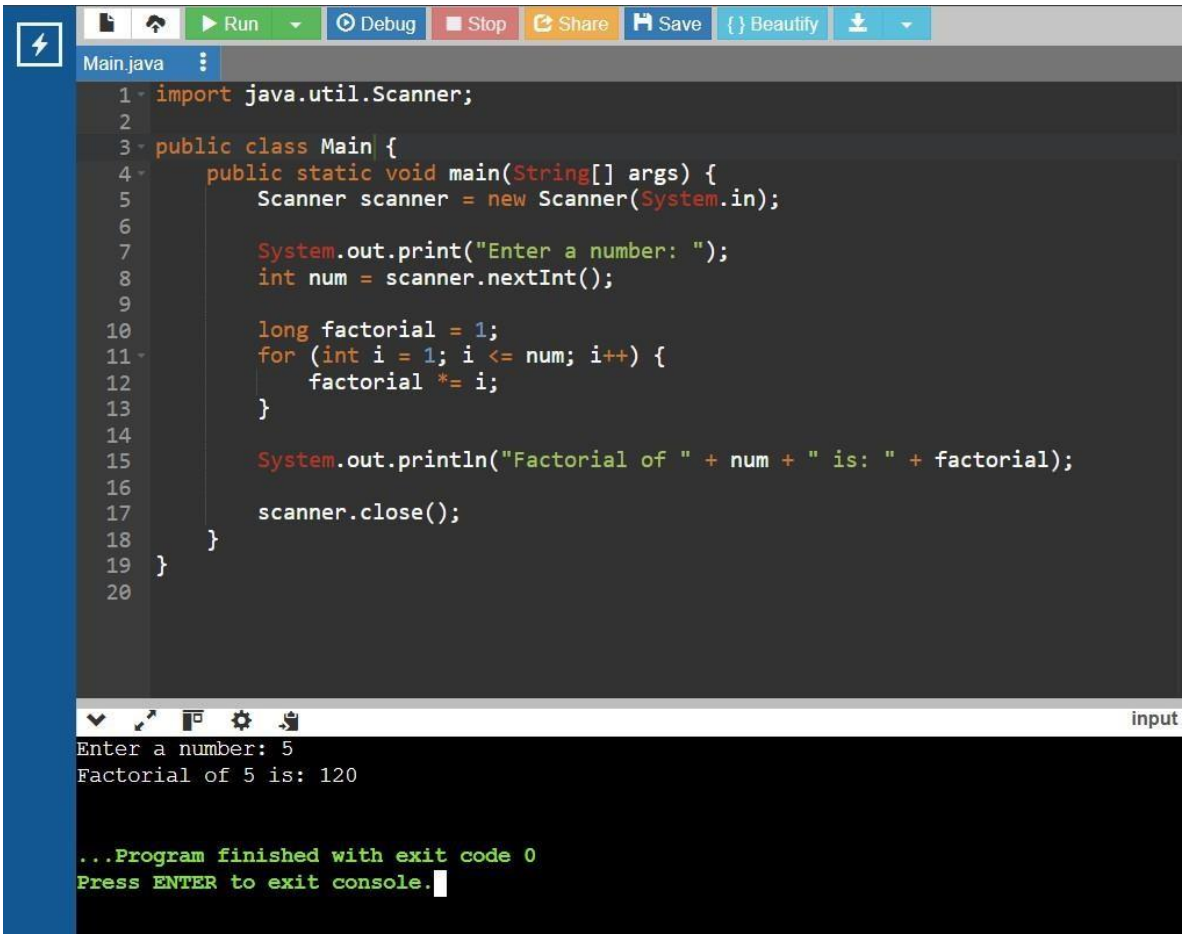
JAVA CODE:-

```
import java.util.Scanner;


public class Main {

   public static void main(String[] args) {

      Scanner scanner = new Scanner(System.in);
```

```
System.out.print("Enter a number: ");
int num = scanner.nextInt();
```

```
        long factorial = 1;
        for (int i = 1; i <= num; i++) {
            factorial *= i;
        }

        System.out.println("Factorial of " + num + " is: " + factorial);

        scanner.close();
    }
}
```

SCREENSHOT WITH OUTPUT:-



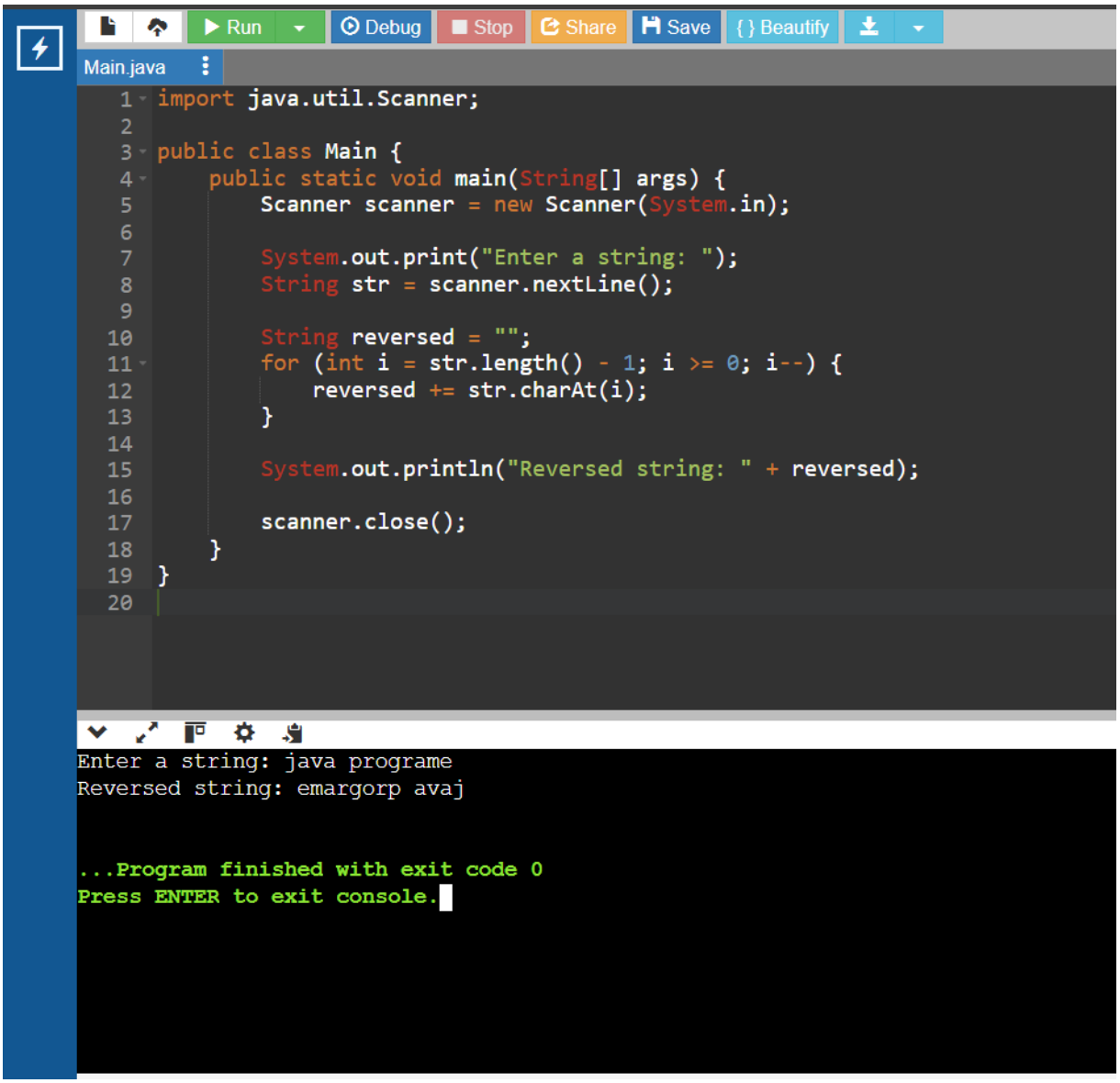
4.h. JAVA PROGRAME FOR REVERSING A STRING:-

AIM:- To Reverse any type of string like names or Anything.

JAVA CODE:-

```
import java.util.Scanner;


public class Main {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);


      System.out.print("Enter a string: ");
      String str = scanner.nextLine();


      String reversed = "";
      for (int i = str.length() - 1; i >= 0; i--) {
         reversed += str.charAt(i);
      }


      System.out.println("Reversed string: " + reversed);


      scanner.close();
   }
}
```

SCREENSHOT WITH OUTPUT:-



4.i. JAVA PROGRAME TO IDENTIFY PRIME NUMBER:-

AIM:- To Identify weather the given number Is prime number or not.

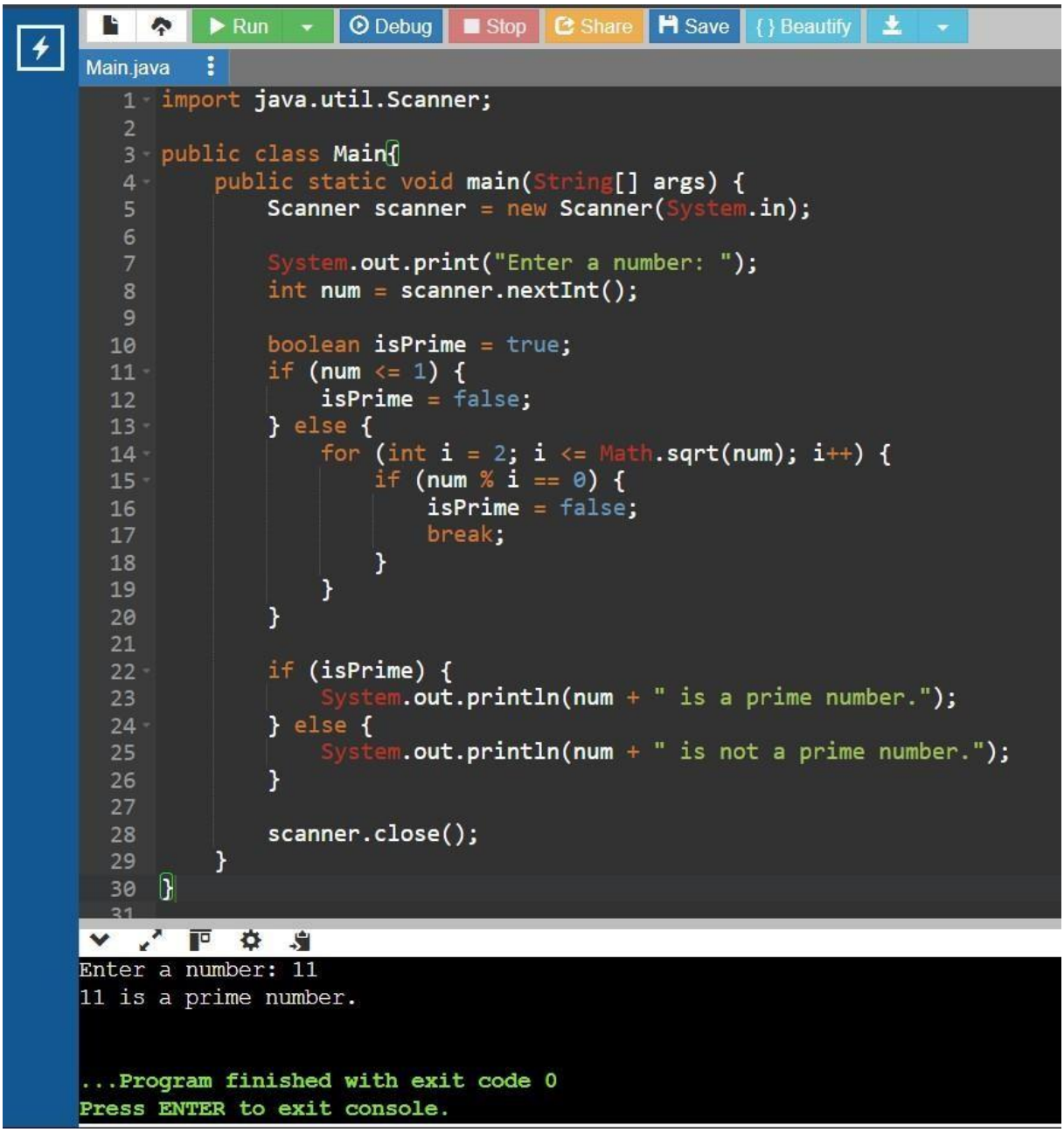
JAVA CODE:-

```
import java.util.Scanner;


public class Main{

    public static void main(String[] args) {
```

```
Scanner scanner = new Scanner(System.in);


System.out.print("Enter a number: ");
```

```
        int num = scanner.nextInt();

        boolean isPrime = true;
        if (num <= 1) {
            isPrime = false;
        } else {
            for (int i = 2; i <= Math.sqrt(num); i++) {
                if (num % i == 0) {
                    isPrime = false;
                    break;
                }
            }
        }

        if (isPrime) {
            System.out.println(num + " is a prime number.");
        } else {
            System.out.println(num + " is not a prime number.");
        }

        scanner.close();
    }
}
```

SCREENSHOT WITH OUTPUT:-



4.j. JAVA PROGRAME FOR PALINDROME:-

AIM:- To find the palindrome of a given string.

JAVA CODE:-

```
import java.util.Scanner;
```

```
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter a word: ");
        String word = scanner.nextLine();

        if (isPalindrome(word)) {
            System.out.println(word + " is a palindrome.");
        } else {
            System.out.println(word + " is not a palindrome.");
        }

        scanner.close();
    }

    public static boolean isPalindrome(String str) {
        int left = 0;
        int right = str.length() - 1;

        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
```

SCREENSHOT OF CODE:-

Run Debug Stop Share Save { } Beautify

Main.java

```java
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a word: ");
        String word = scanner.nextLine();

        if (isPalindrome(word)) {
            System.out.println(word + " is a palindrome.");
        } else {
            System.out.println(word + " is not a palindrome.");
        }

        scanner.close();
    }

    public static boolean isPalindrome(String str) {
        int left = 0;
        int right = str.length() - 1;

        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
```

OUTPUT:-

```
Enter a word: ABCDCBA
ABCDCBA is a palindrome.


...Program finished with exit code 0
Press ENTER to exit console.
```

Inheritence:-

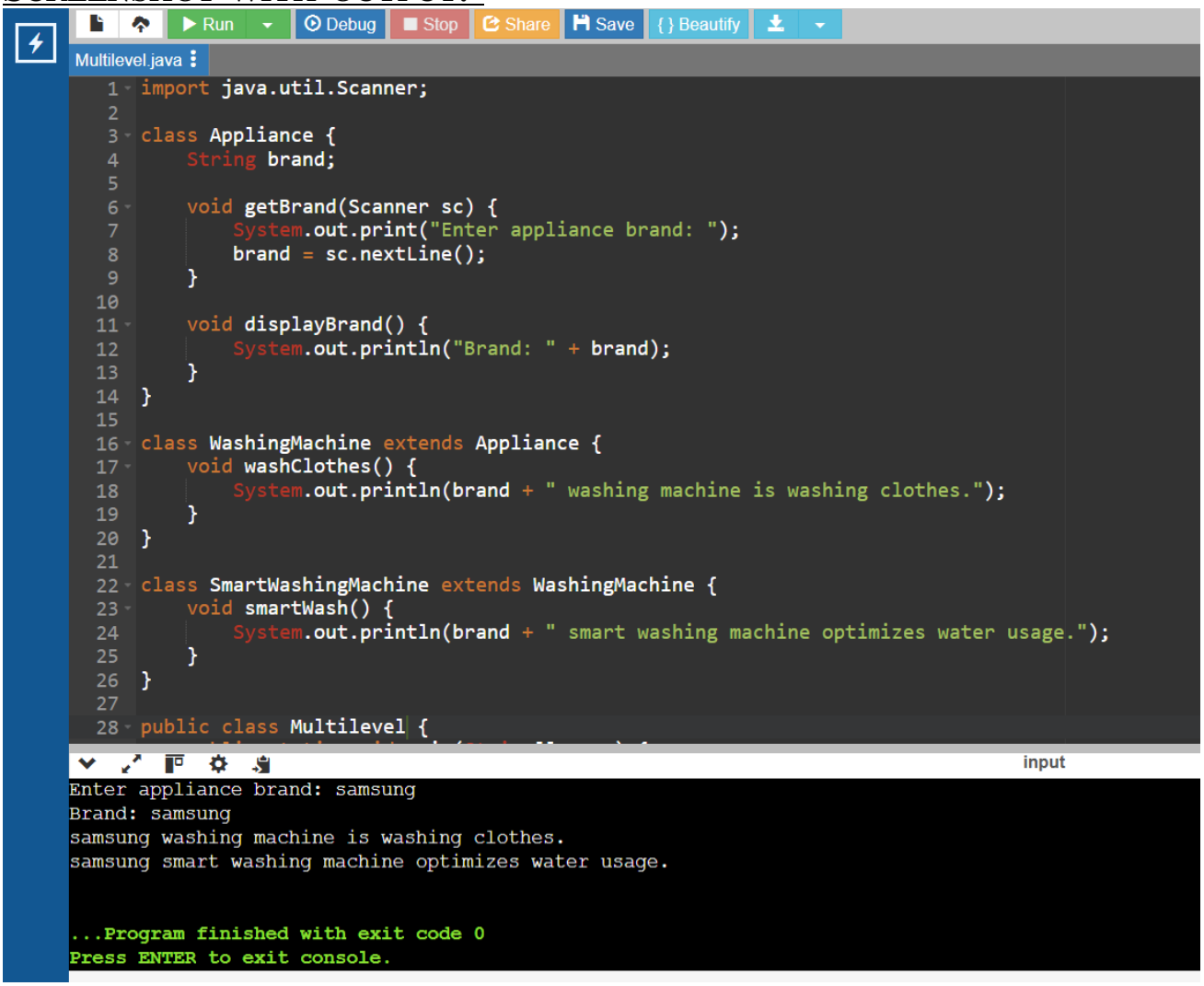
5.Multilevel

a)

```
import java.util.Scanner;

class Appliance {
   String brand;

   void getBrand(Scanner sc) {
      System.out.print("Enter appliance brand: ");
      brand = sc.nextLine();
   }

   void displayBrand() {
      System.out.println("Brand: " + brand);
   }
}

class WashingMachine extends Appliance {
   void washClothes() {
      System.out.println(brand + " washing machine is washing clothes.");
   }
}

class SmartWashingMachine extends WashingMachine {
   void smartWash() {
      System.out.println(brand + " smart washing machine optimizes water usage.");
   }
}

public class Multilevel {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      SmartWashingMachine machine = new SmartWashingMachine();
      machine.getBrand(sc);
      machine.displayBrand();
      machine.washClothes();
      machine.smartWash();
```

```
        sc.close();
    }
}
```

SCREENSHOT WITH OUTPUT:-



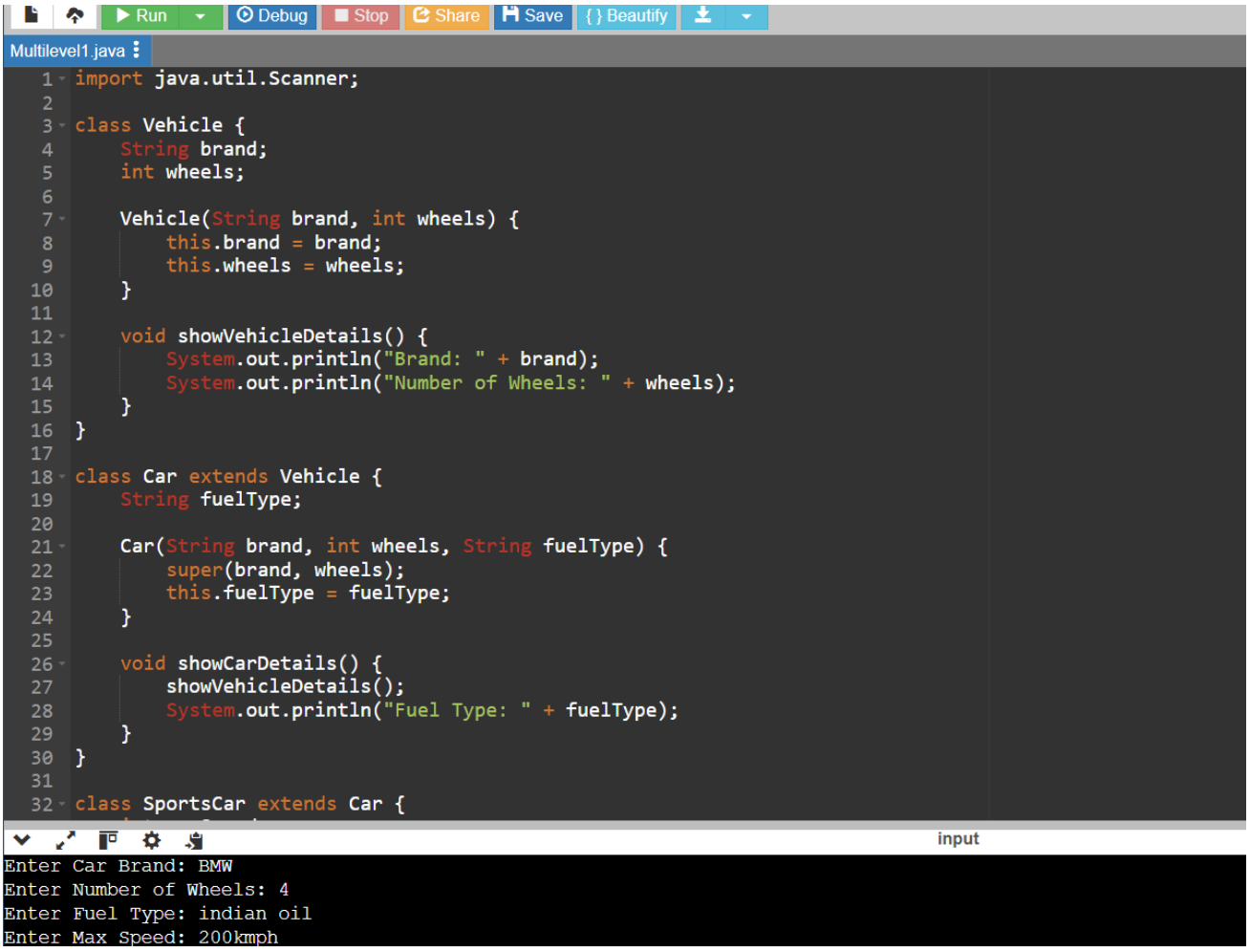
b)

```
import java.util.Scanner;

class Vehicle {
   String brand;
   int wheels;

   Vehicle(String brand, int wheels) {
      this.brand = brand;
      this.wheels = wheels;
   }
```

```
    void showVehicleDetails() {
        System.out.println("Brand: " + brand);
        System.out.println("Number of Wheels: " + wheels);
    }
}

class Car extends Vehicle {
    String fuelType;

    Car(String brand, int wheels, String fuelType) {
        super(brand, wheels);
        this.fuelType = fuelType;
    }

    void showCarDetails() {
        showVehicleDetails();
        System.out.println("Fuel Type: " + fuelType);
    }
}

class SportsCar extends Car {
    int maxSpeed;

    SportsCar(String brand, int wheels, String fuelType, int maxSpeed) {
        super(brand, wheels, fuelType);
        this.maxSpeed = maxSpeed;
    }

    void showSportsCarDetails() {
        showCarDetails();
        System.out.println("Max Speed: " + maxSpeed + " km/h");
    }
}

public class Multilevel1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter Car Brand: ");
        String brand = sc.nextLine();
        System.out.print("Enter Number of Wheels: ");
        int wheels = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter Fuel Type: ");
        String fuelType = sc.nextLine();
        System.out.print("Enter Max Speed: ");
        int maxSpeed = sc.nextInt();

        SportsCar sportsCar = new SportsCar(brand, wheels, fuelType, maxSpeed);
```

```
        System.out.println("\nSports Car Details:");
        sportsCar.showSportsCarDetails();

        sc.close();
    }
}
```

SCREENSHOT WITH OUTPUT:-



6.Hybrid:-

a)

```
class Vehicle {
    void start() {
        System.out.println("Vehicle is starting...");
```

```
    }
}


interface ElectricVehicle {
    void chargeBattery();
}


interface FuelVehicle {
    void refuel();
}


class Tesla extends Vehicle implements ElectricVehicle {
    public void chargeBattery() {
        System.out.println("Tesla is charging its battery.");
    }

    void autopilot() {
        System.out.println("Tesla is in autopilot mode.");
    }
}


class Ford extends Vehicle implements FuelVehicle {
    public void refuel() {
        System.out.println("Ford is refueling with gasoline.");
    }

    void manualDrive() {
        System.out.println("Ford is being driven manually.");
    }
}

public class Hybrid {
    public static void main(String[] args) {
        Tesla myTesla = new Tesla();
        myTesla.start();        // Inherited from Vehicle
        myTesla.chargeBattery(); // Implemented from ElectricVehicle
        myTesla.autopilot();     // Specific to Tesla

        System.out.println("-----------------");

        Ford myFord = new Ford();
```

```
        myFord.start();   // Inherited from Vehicle
        myFord.refuel();  // Implemented from FuelVehicle
        myFord.manualDrive();  // Specific to Ford
    }
     }
```

SCREENSHOT WITH OUTPUT:-

Run Debug Stop Share Save { } Beautify

Hybrid.java

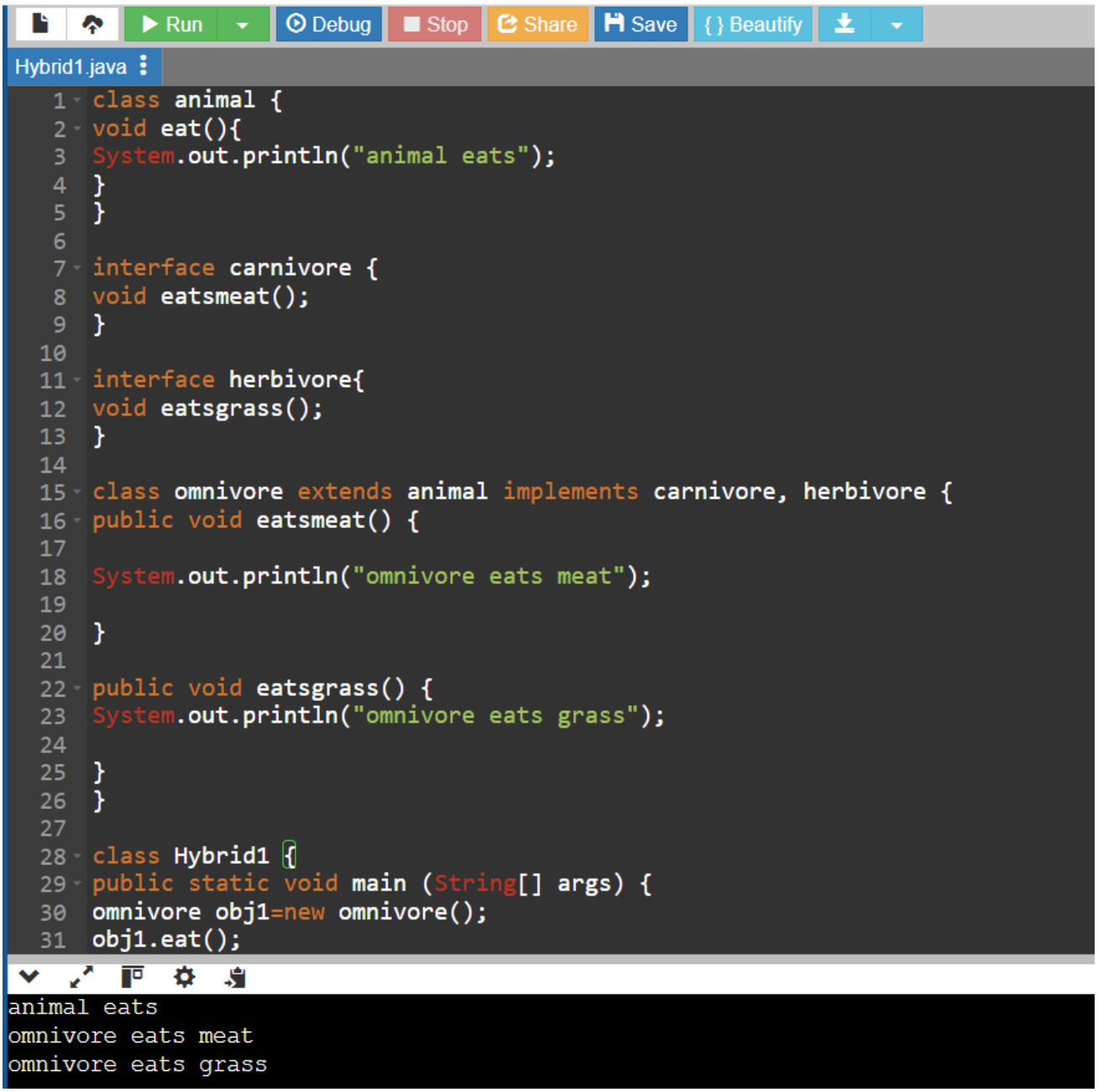
```java
class Vehicle {
    void start() {
        System.out.println("Vehicle is starting...");
    }
}


interface ElectricVehicle {
    void chargeBattery();
}


interface FuelVehicle {
    void refuel();
}


class Tesla extends Vehicle implements ElectricVehicle {
    public void chargeBattery() {
        System.out.println("Tesla is charging its battery.");
    }

    void autopilot() {
        System.out.println("Tesla is in autopilot mode.");
    }
}


class Ford extends Vehicle implements FuelVehicle {
    public void refuel() {
```

```
Vehicle is starting...
Tesla is charging its battery.
Tesla is in autopilot mode.
----------------
Vehicle is starting...
Ford is refueling with gasoline.
```

b)

```
class animal {
void eat(){
System.out.println("animal eats");
}
}

interface carnivore {
void eatsmeat();
}

interface herbivore{
void eatsgrass();
}

class omnivore extends animal implements carnivore, herbivore {
public void eatsmeat() {

System.out.println("omnivore eats meat");

}

public void eatsgrass() {
System.out.println("omnivore eats grass");

}
}

class Hybrid1 {
public static void main (String[] args) {
omnivore obj1=new omnivore();
obj1.eat();
obj1.eatsmeat();
obj1.eatsgrass();
}

}
```

SCREENSHOT WITH OUTPUT:-



7.single Inheritence:-

a)

```
class Employee {
String name;
double salary;

void setDetails(String name, double salary) {
   this.name = name;
   this.salary = salary;
```

```
    }

    void showDetails() {
        System.out.println("Employee Name: " + name);
        System.out.println("Salary: $" + salary);
    }
}


class Manager extends Employee {
    String department;

    void setDepartment(String department) {
        this.department = department;
    }

    void showManagerDetails() {
        showDetails(); // Call parent class method
        System.out.println("Department: " + department);
    }
}


public class Single {
    public static void main(String[] args) {
        Manager m = new Manager();
        m.setDetails("Alice Johnson", 75000);
        m.setDepartment("IT");

        System.out.println("Manager Details:");
        m.showManagerDetails();
    }
     }
```

SCREENSHOT WITH OUTPUT:-

Run Debug Stop Share Save { } Beautify

Single.java

```java
class Employee {
    String name;
    double salary;

    void setDetails(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }

    void showDetails() {
        System.out.println("Employee Name: " + name);
        System.out.println("Salary: $" + salary);
    }
}


class Manager extends Employee {
    String department;

    void setDepartment(String department) {
        this.department = department;
    }

    void showManagerDetails() {
        showDetails(); // Call parent class method
        System.out.println("Department: " + department);
    }
}
```

```
Manager Details:
Employee Name: Alice Johnson
Salary: $75000.0
Department: IT


...Program finished with exit code 0
Press ENTER to exit console.
```

b)

class Person {

```
    String name;
    int age;

    void display() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}

class Student extends Person {
    int studentId;

    void showStudentInfo() {
        System.out.println("Student ID: " + studentId);
    }
}

public class Single1 {
    public static void main(String[] args) {
        Student s = new Student();
        s.name = "Alice";
        s.age = 20;
        s.studentId = 101;
        s.display();
        s.showStudentInfo();
    }
}
```

SCREENSHOT WITH OUTPUT:-

Run Debug Stop Share Save { } Beautify

Single1.java

```java
class Person {
    String name;
    int age;

    void display() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}

class Student extends Person {
    int studentId;

    void showStudentInfo() {
        System.out.println("Student ID: " + studentId);
    }
}

public class Single1 {
    public static void main(String[] args) {
        Student s = new Student();
        s.name = "Alice";
        s.age = 20;
        s.studentId = 101;
        s.display();
        s.showStudentInfo();
    }
}
```

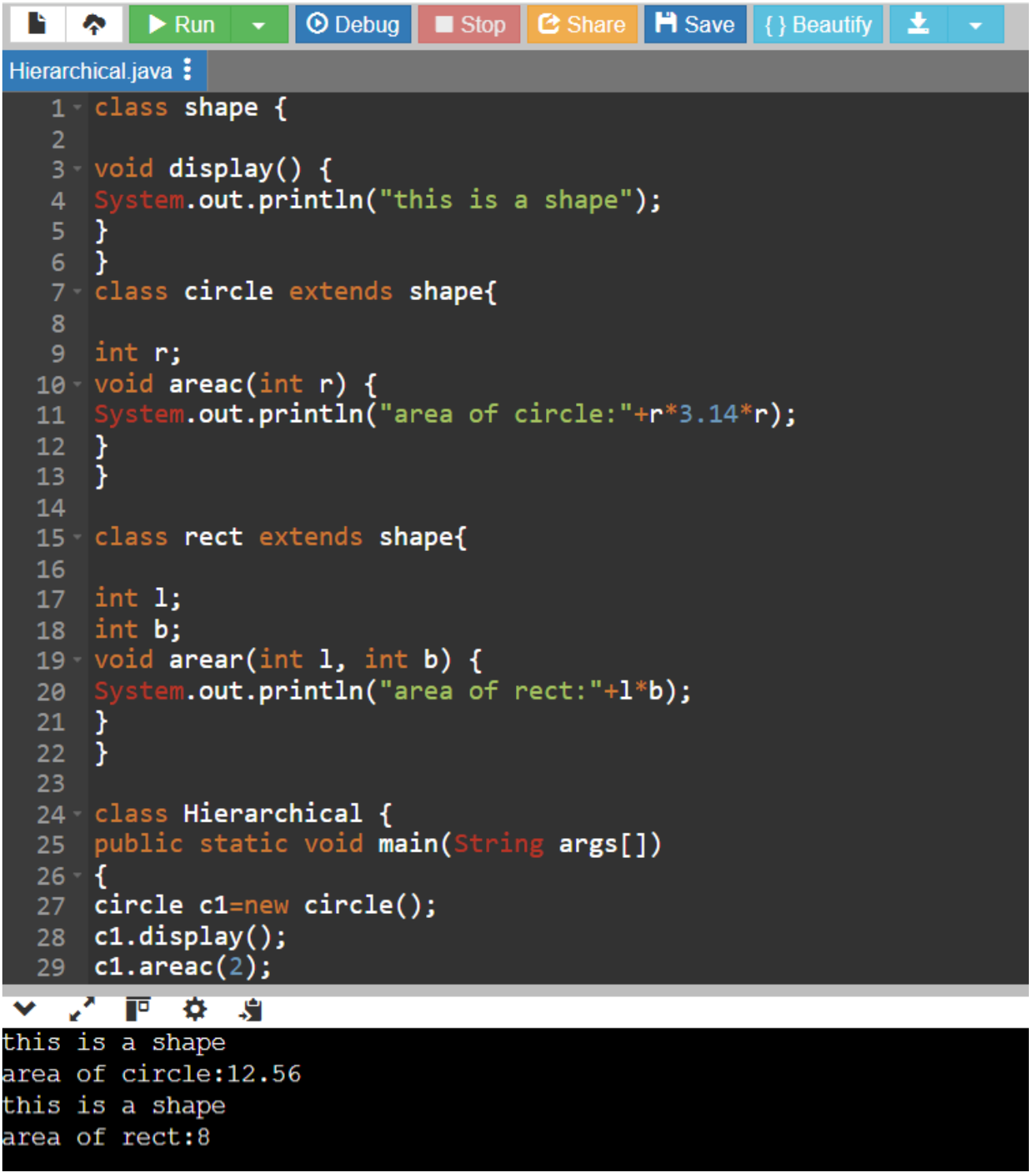
```
Name: Alice, Age: 20
Student ID: 101


...Program finished with exit code 0
Press ENTER to exit console.
```

8.Hierarchical Inheritence:-

a)

```
class shape {

void display() {
System.out.println("this is a shape");
}
}
class circle extends shape{

int r;
void areac(int r) {
System.out.println("area of circle:"+r*3.14*r);
}
}

class rect extends shape{

int l;
int b;
void arear(int l, int b) {
System.out.println("area of rect:"+l*b);
}
}

class Hierarchical {
public static void main(String args[])
{
circle c1=new circle();
c1.display();
c1.areac(2);
rect r1=new rect();
r1.display();
r1.arear(4,2);
}}
```

SCREENSHOT WITH OUTPUT:-



b)

```
import java.util.*;
class person{
   String name;
   int age;
```

```
    String address;

    person(String name,int age, String address){
        this.name=name;
        this.age=age;
        this.address=address;
    }

    void displayInfo(){
        System.out.println("Name - "+name);
        System.out.println("age - "+age);
        System.out.println("address - "+address);
    }
}

class student extends person{
    String[] course = new String[5];
    int id;

    student(String name,int age, String address,String[] course,int id){
        super(name,age,address);
        this.course=course;
        this.id=id;
    }
    void displayStudentInfo(){
        for(String name:course){
            System.out.print(name+", ");

        }
        System.out.println();
        System.out.println("Id - "+id);
        displayInfo();
    }


}

class teacher extends person{
    String subject;

    teacher(String name,int age, String address,String subject){
        super(name,age,address);
        this.subject=subject;

    }
```

```
    void displayTeacherInfo(){
        System.out.println("subject - "+subject);
        displayInfo();
    }


}

public class Hierarchical1 {
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        String[] arr= new String[4];
        for (int i=0 ; i<=3 ; i++){
            String arrele=sc.nextLine();
            arr[i]=arrele;
        }
        String name = sc.nextLine();
        String address = sc.nextLine();
        int age = sc.nextInt();
        int id = sc.nextInt();
        student obj1 = new student(name,age,address,arr,id);
        System.out.println("------------Students details-----------");
        obj1.displayStudentInfo();


        sc.nextLine();
        String tname = sc.nextLine();
        String taddress = sc.nextLine();
        String tsubject = sc.nextLine();
        int tage = sc.nextInt();
        teacher obj2 = new teacher(tname,tage,taddress,tsubject);
        System.out.println("------------Teacher details-----------");
        obj2.displayTeacherInfo();
    }
}
```

SCREENSHOT WITH OUTPUT:-

Run Debug Stop Share Save { } Beautify

Hierarchical1.java

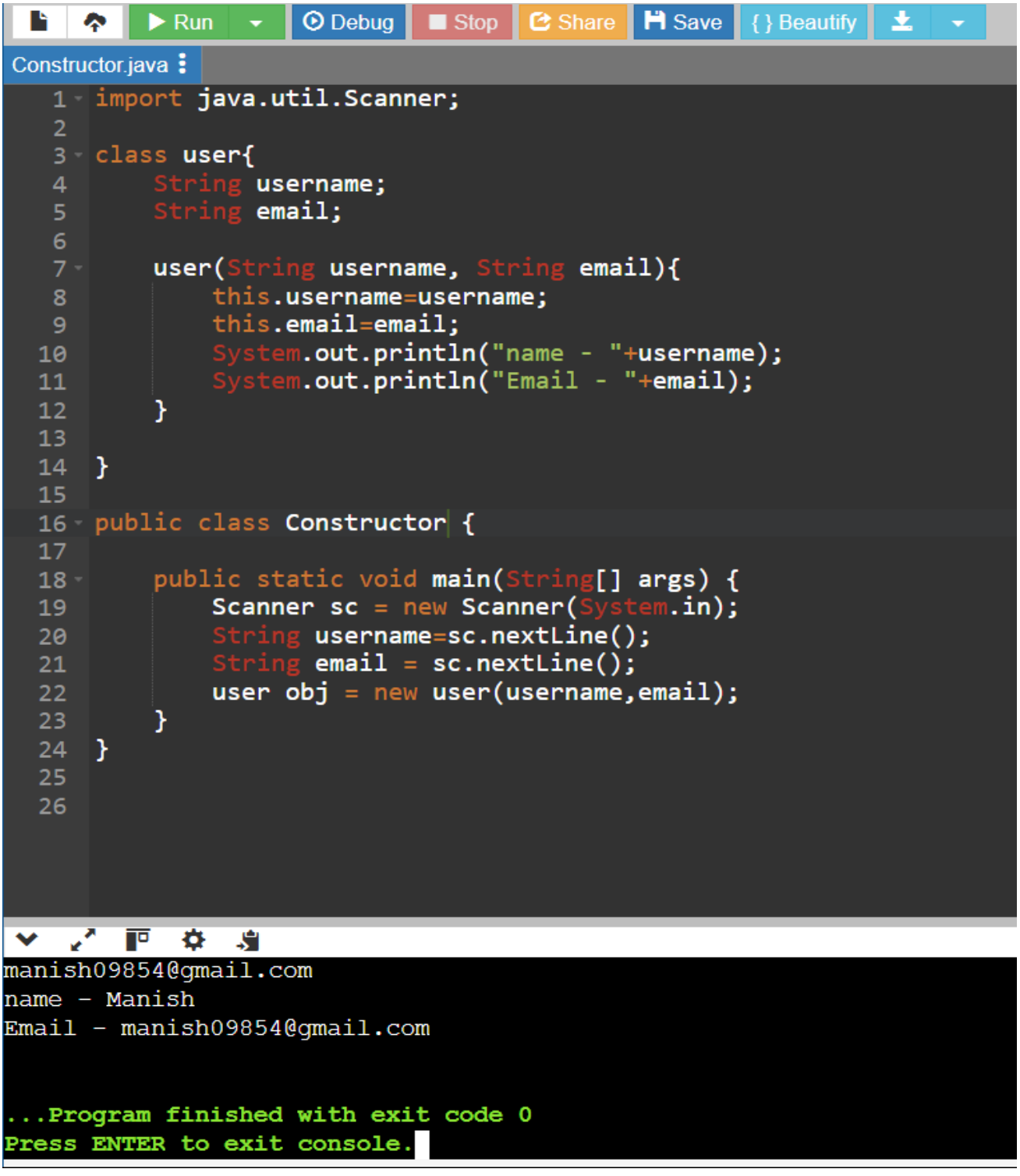
```java
import java.util.*;
class person{
    String name;
    int age;
    String address;

    person(String name,int age, String address){
        this.name=name;
        this.age=age;
        this.address=address;
    }

    void displayInfo(){
        System.out.println("Name - "+name);
        System.out.println("age - "+age);
        System.out.println("address - "+address);
    }
}

class student extends person{
    String[] course = new String[5];
    int id;

    student(String name,int age, String address,String[] course,int id){
        super(name,age,address);
        this.course=course;
        this.id=id;
    }
    void displayStudentInfo(){
        for(String name:course){
            System.out.print(name+", ");
```

```
Manish
18
kadapa
```

9.Constructor:-

```
import java.util.Scanner;

class user{
   String username;
   String email;

   user(String username, String email){
      this.username=username;
      this.email=email;
      System.out.println("name - "+username);
      System.out.println("Email - "+email);
   }

}

public class Constructor {

   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String username=sc.nextLine();
      String email = sc.nextLine();
      user obj = new user(username,email);
   }
}
```

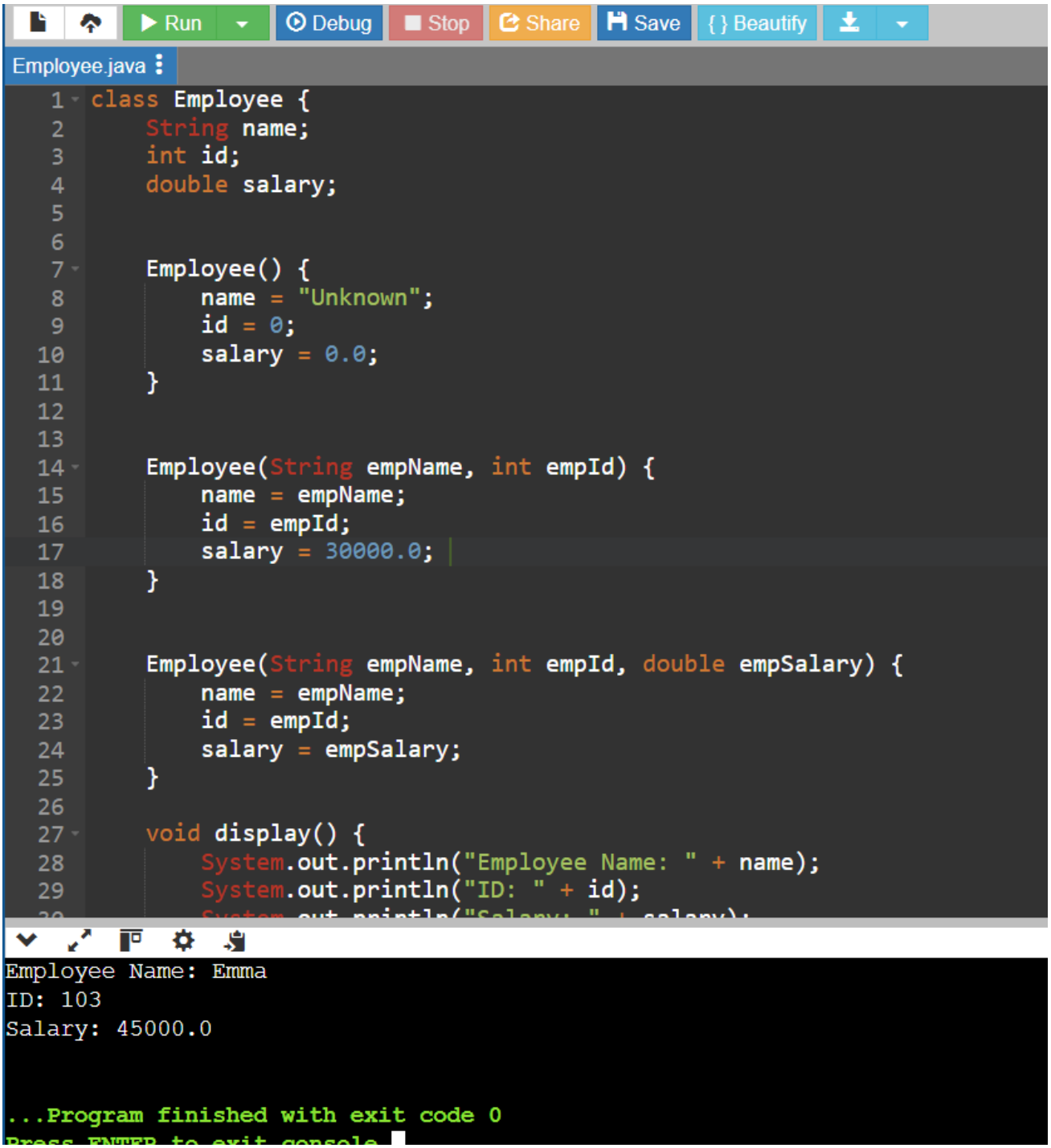
SCREENSHOT WITH OUTPUT:-



10.Constructor overloading:-

```
class Employee {
   String name;
   int id;
   double salary;


   Employee() {
      name = "Unknown";
      id = 0;
      salary = 0.0;
   }


   Employee(String empName, int empId) {
      name = empName;
      id = empId;
      salary = 30000.0;
   }


   Employee(String empName, int empId, double empSalary) {
      name = empName;
      id = empId;
      salary = empSalary;
   }

   void display() {
      System.out.println("Employee Name: " + name);
      System.out.println("ID: " + id);
      System.out.println("Salary: " + salary);
   }

   public static void main(String[] args) {
      Employee e1 = new Employee();
      Employee e2 = new Employee("John", 102);
      Employee e3 = new Employee("Emma", 103, 45000.0);

      e1.display();
      e2.display();
      e3.display();
   }
}
```

SCREENSHOT WITH OUTPUT:-

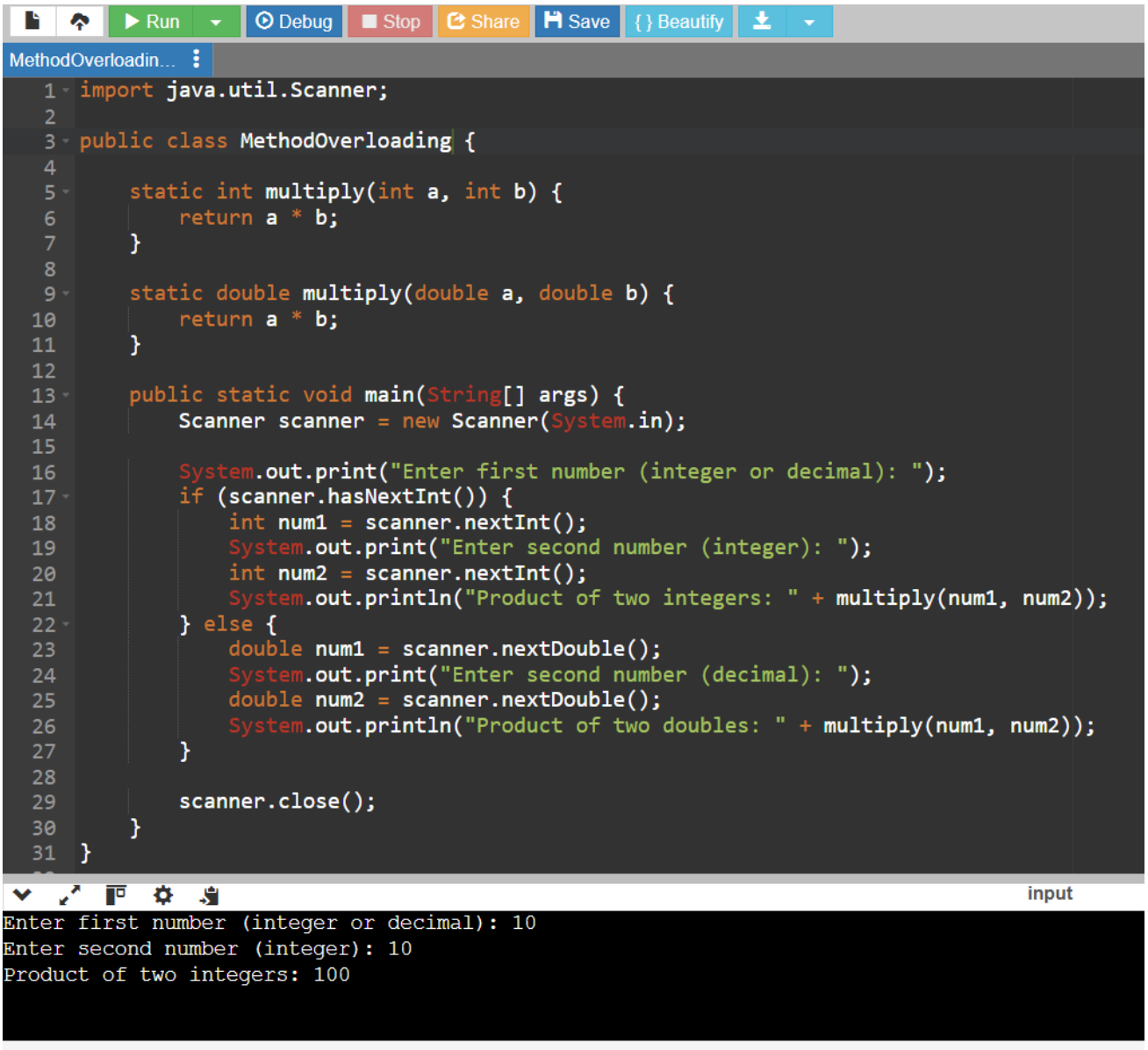


11.Method overloading:-

a)

```
import java.util.Scanner;

public class MethodOverloading {

   static int multiply(int a, int b) {
      return a * b;
   }

   static double multiply(double a, double b) {
      return a * b;
   }

   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      System.out.print("Enter first number (integer or decimal): ");
      if (scanner.hasNextInt()) {
         int num1 = scanner.nextInt();
         System.out.print("Enter second number (integer): ");
         int num2 = scanner.nextInt();
         System.out.println("Product of two integers: " + multiply(num1, num2));
      } else {
         double num1 = scanner.nextDouble();
         System.out.print("Enter second number (decimal): ");
         double num2 = scanner.nextDouble();
         System.out.println("Product of two doubles: " + multiply(num1, num2));
      }

      scanner.close();
   }
}
```

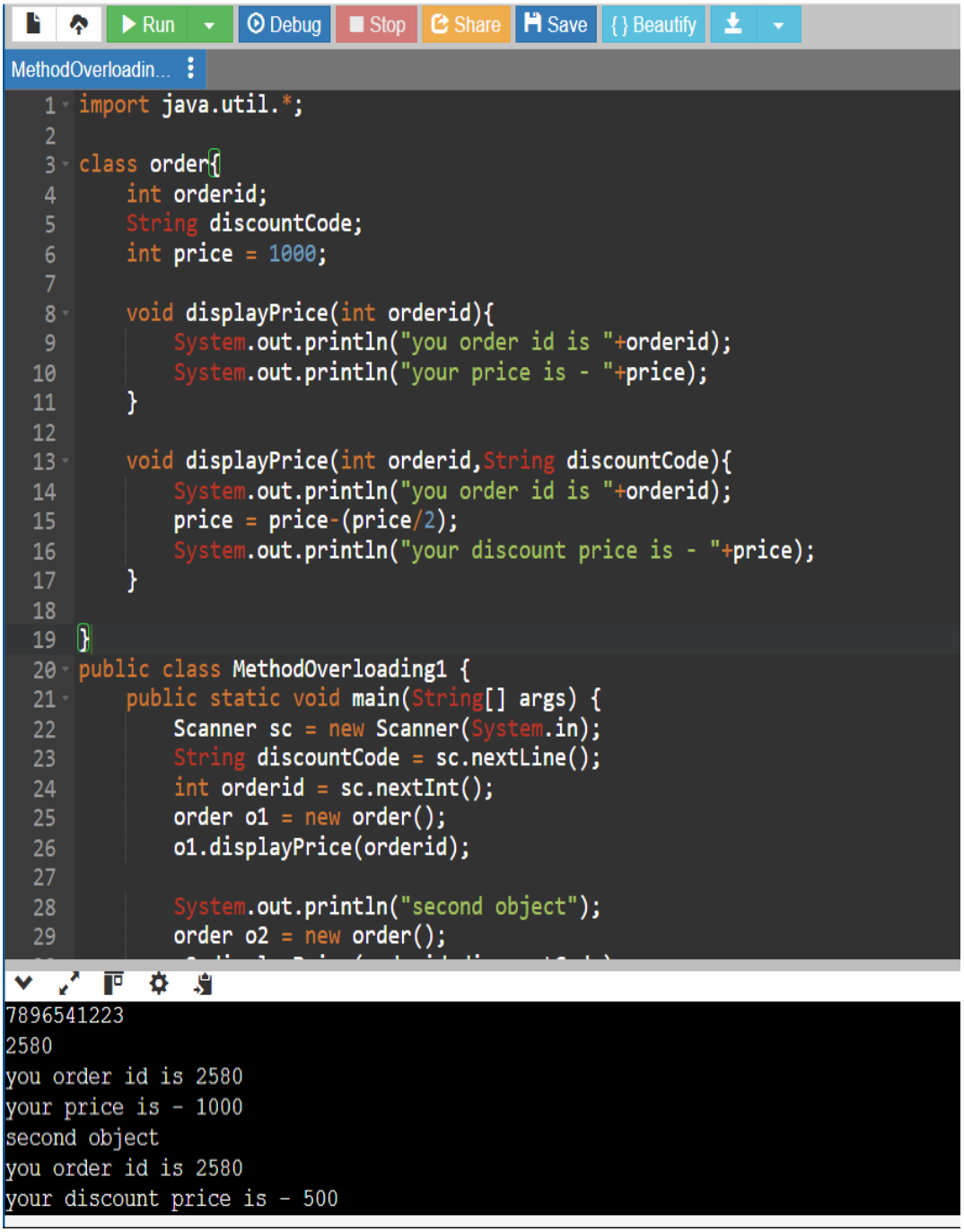
SCREENSHOT WITH OUTPUT:-



b)

```
import java.util.*;

class order{
   int orderid;
   String discountCode;
   int price = 1000;

   void displayPrice(int orderid){
      System.out.println("you order id is "+orderid);
      System.out.println("your price is - "+price);
   }

   void displayPrice(int orderid,String discountCode){
      System.out.println("you order id is "+orderid);
      price = price-(price/2);
      System.out.println("your discount price is - "+price);
   }

}
public class MethodOverloading1 {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String discountCode = sc.nextLine();
      int orderid = sc.nextInt();
      order o1 = new order();
      o1.displayPrice(orderid);

      System.out.println("second object");
      order o2 = new order();
      o2.displayPrice(orderid,discountCode);
      }
}
```

SCREENSHOT WITH OUTPUT:-

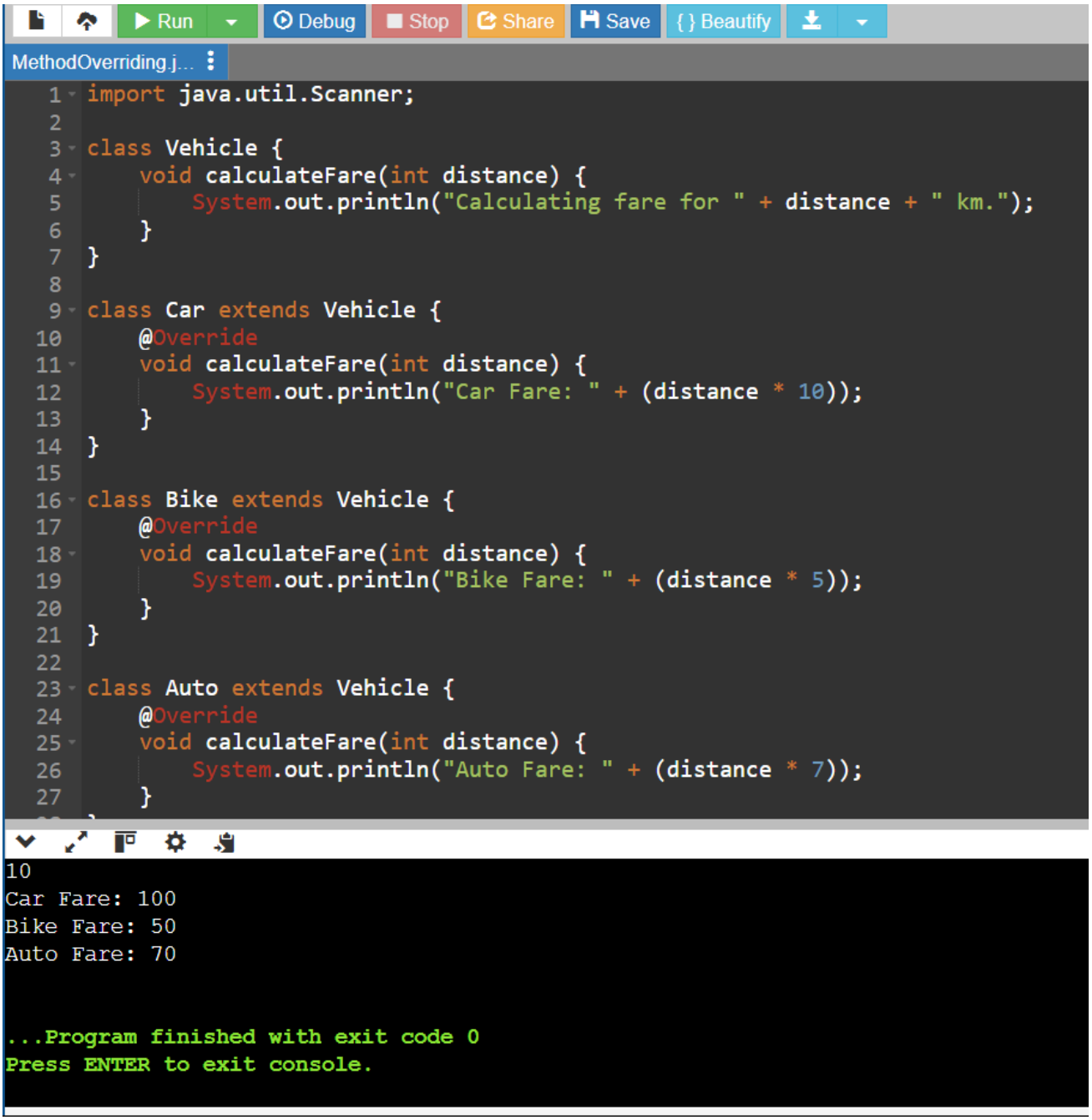


12.Method overriding:-

a)

```
import java.util.Scanner;

class Vehicle {
   void calculateFare(int distance) {
      System.out.println("Calculating fare for " + distance + " km.");
   }
}

class Car extends Vehicle {
   @Override
   void calculateFare(int distance) {
      System.out.println("Car Fare: " + (distance * 10));
   }
}

class Bike extends Vehicle {
   @Override
   void calculateFare(int distance) {
      System.out.println("Bike Fare: " + (distance * 5));
   }
}

class Auto extends Vehicle {
   @Override
   void calculateFare(int distance) {
      System.out.println("Auto Fare: " + (distance * 7));
   }
}

public class MethodOverriding{
   public static void main(String[] args) {
      Vehicle car = new Car();
      Vehicle bike = new Bike();
      Vehicle auto = new Auto();
      Scanner sc = new Scanner(System.in);
      int distance = sc.nextInt();
      car.calculateFare(distance);
      bike.calculateFare(distance);
      auto.calculateFare(distance);
   }
}
```

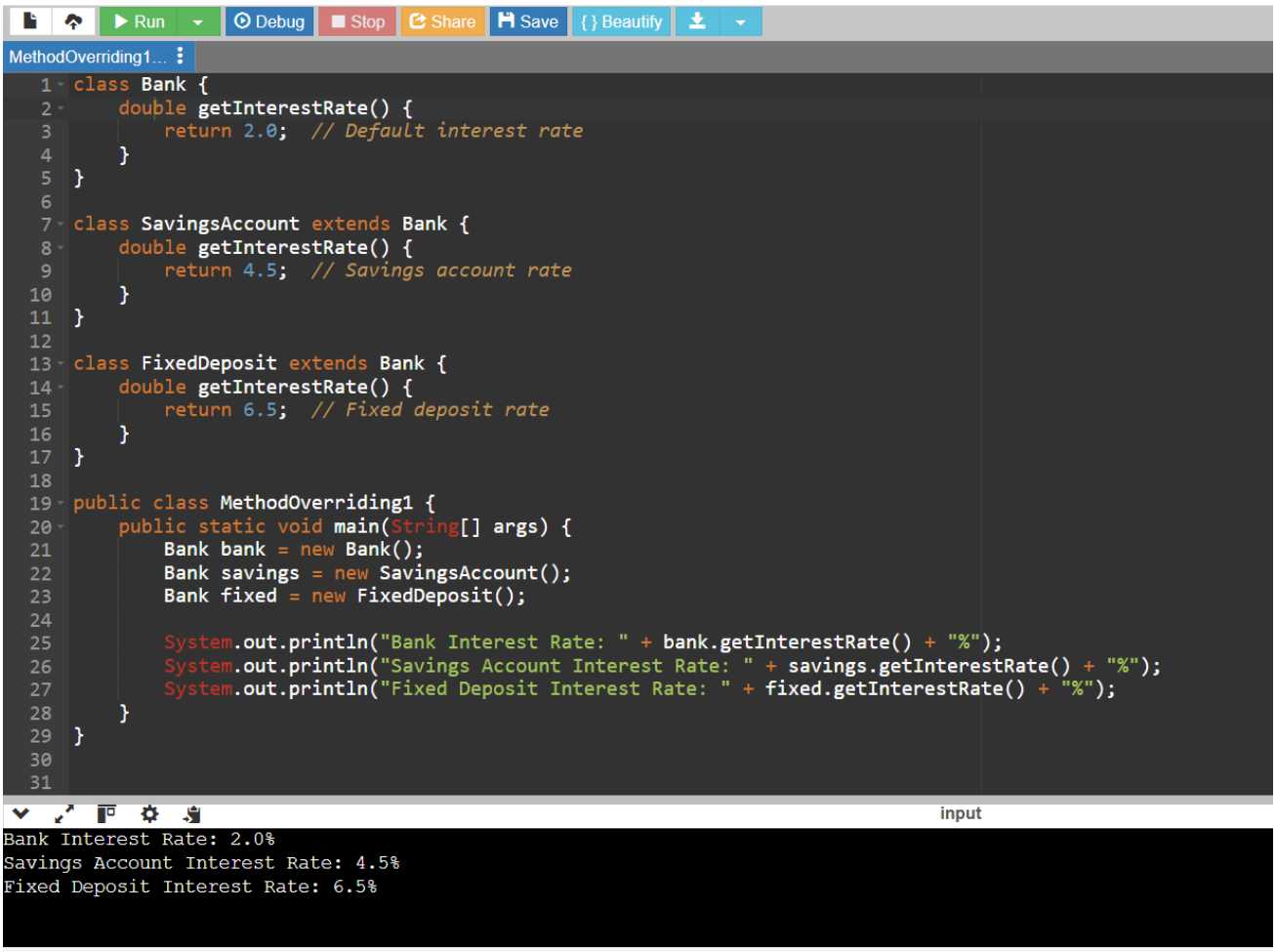
SCREENSHOT WITH OUTPUT:-



b)

```
class Bank {
   double getInterestRate() {
      return 2.0;  // Default interest rate
   }
}

class SavingsAccount extends Bank {
   double getInterestRate() {
      return 4.5;  // Savings account rate
   }
}

class FixedDeposit extends Bank {
   double getInterestRate() {
      return 6.5;  // Fixed deposit rate
   }
}

public class MethodOverriding1 {
   public static void main(String[] args) {
      Bank bank = new Bank();
      Bank savings = new SavingsAccount();
      Bank fixed = new FixedDeposit();

      System.out.println("Bank Interest Rate: " + bank.getInterestRate() + "%");
      System.out.println("Savings Account Interest Rate: " + savings.getInterestRate() + "%");
      System.out.println("Fixed Deposit Interest Rate: " + fixed.getInterestRate() + "%");
   }
}
```

SCREENSHOT WITH OUTPUT:-



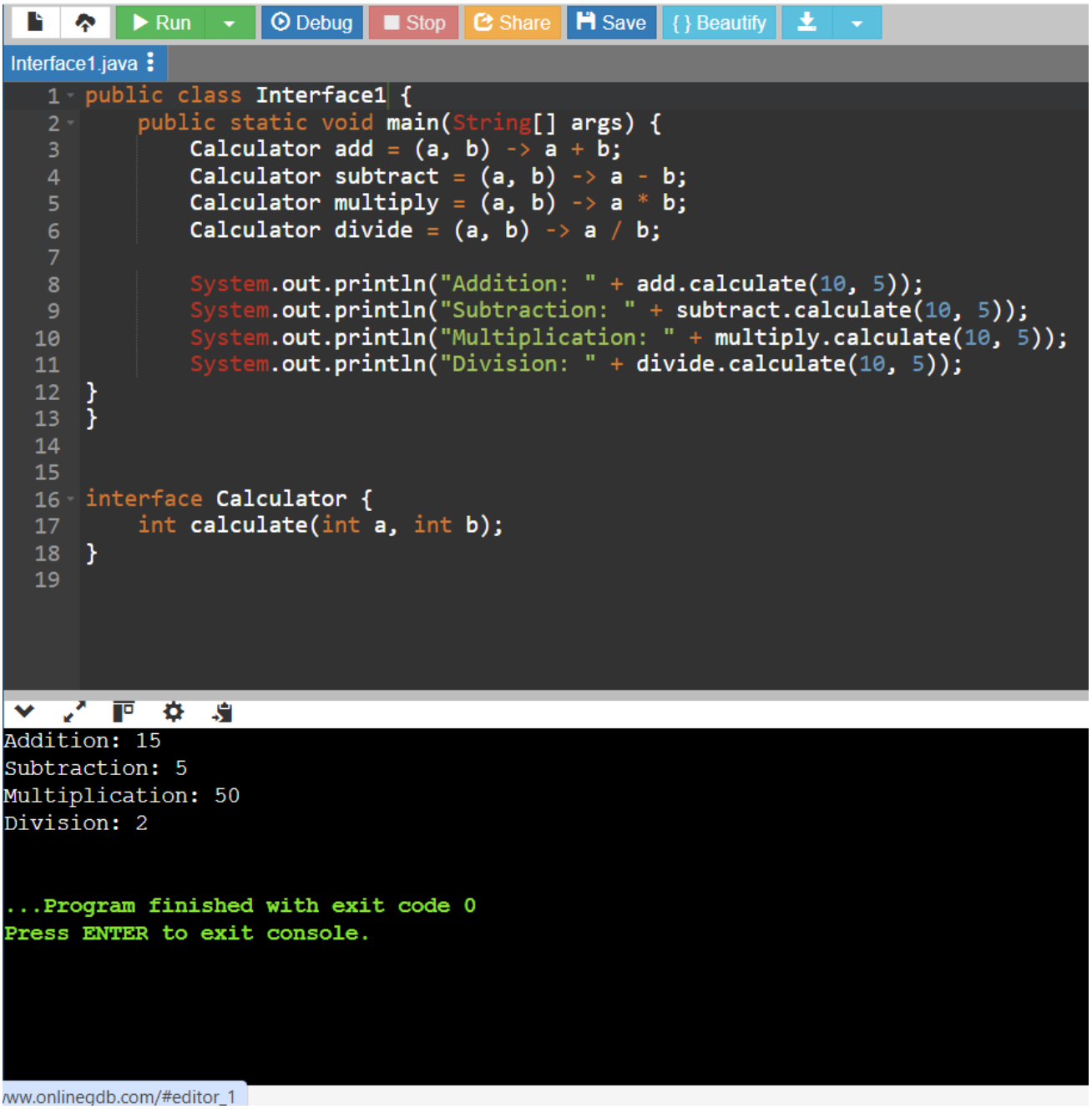
ABSTRACTION

13.Interface Programes:-

a)

```
public class Interface1 {
   public static void main(String[] args) {
      Calculator add = (a, b) -> a + b;
      Calculator subtract = (a, b) -> a - b;
      Calculator multiply = (a, b) -> a * b;
      Calculator divide = (a, b) -> a / b;

      System.out.println("Addition: " + add.calculate(10, 5));
      System.out.println("Subtraction: " + subtract.calculate(10, 5));
      System.out.println("Multiplication: " + multiply.calculate(10, 5));
      System.out.println("Division: " + divide.calculate(10, 5));
}
}


interface Calculator {
   int calculate(int a, int b);
    }
```

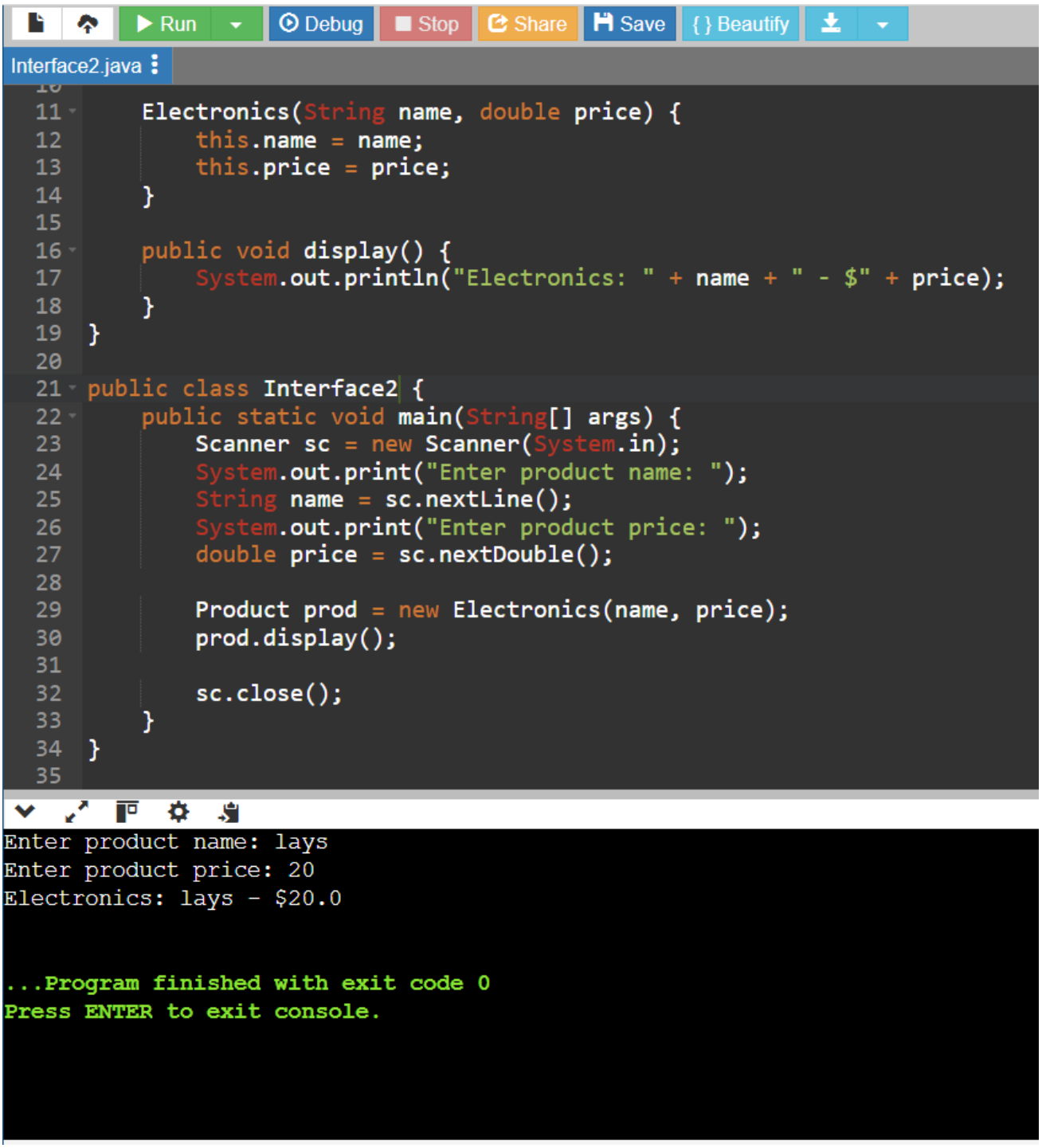
SCREENSHOT WITH OUTPUT:-



b)

```
import java.util.Scanner;

interface Product {
   void display();
}

class Electronics implements Product {
   String name;
   double price;

   Electronics(String name, double price) {
      this.name = name;
      this.price = price;
   }

   public void display() {
      System.out.println("Electronics: " + name + " - $" + price);
   }
}

public class Interface2 {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter product name: ");
      String name = sc.nextLine();
      System.out.print("Enter product price: ");
      double price = sc.nextDouble();

      Product prod = new Electronics(name, price);
      prod.display();

      sc.close();
   }
}
```

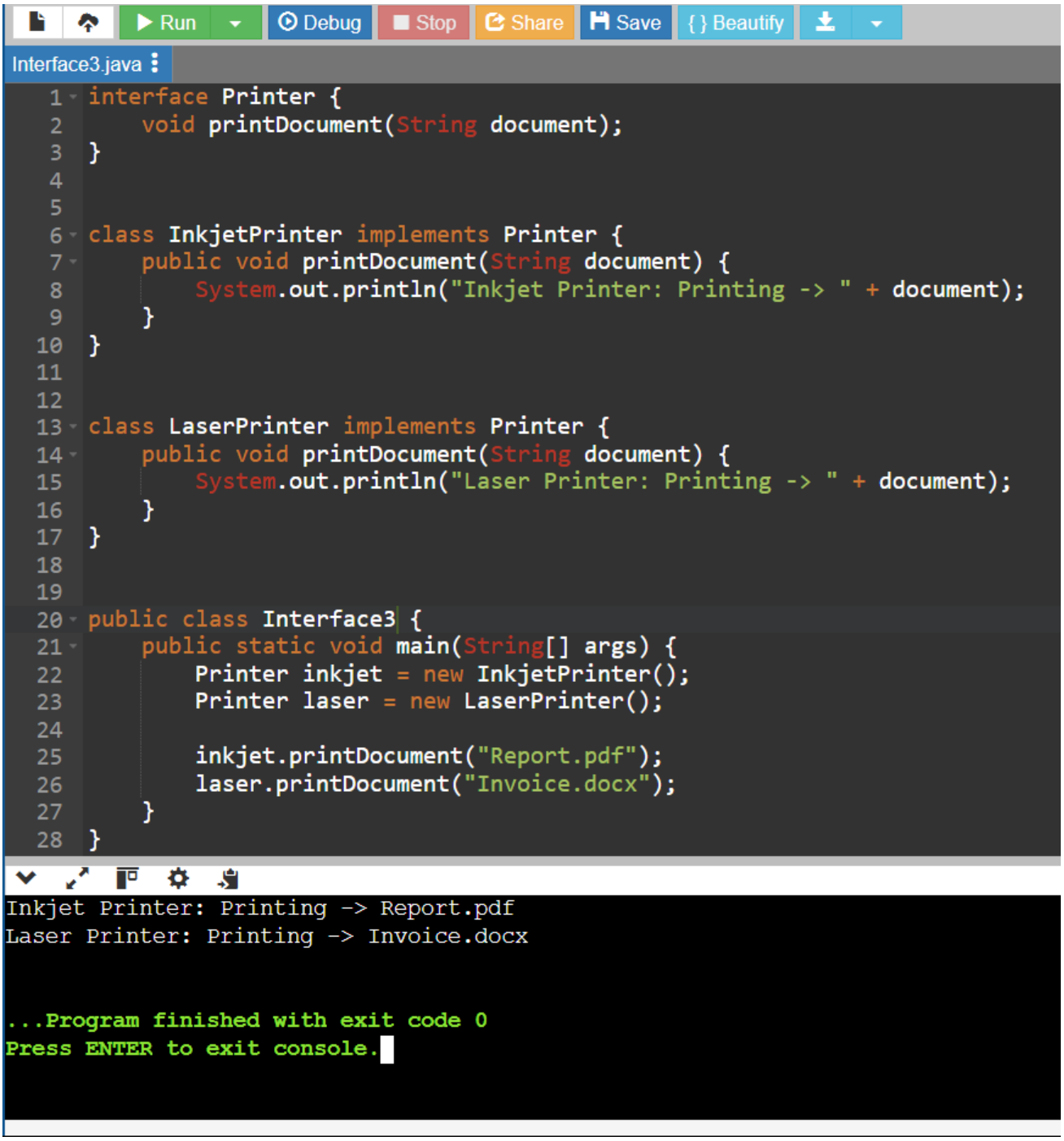
SCREENSHOT WITH OUTPUT:-



c)

```
interface Printer {
   void printDocument(String document);
}


class InkjetPrinter implements Printer {
   public void printDocument(String document) {
      System.out.println("Inkjet Printer: Printing -> " + document);
   }
}


class LaserPrinter implements Printer {
   public void printDocument(String document) {
      System.out.println("Laser Printer: Printing -> " + document);
   }
}


public class Interface3 {
   public static void main(String[] args) {
      Printer inkjet = new InkjetPrinter();
      Printer laser = new LaserPrinter();

      inkjet.printDocument("Report.pdf");
      laser.printDocument("Invoice.docx");
   }
}
```

SCREENSHOT WITH OUTPUT:-



d)

```
interface a {
void methoda();
}

interface b extends a {
void methodb();
}

class c implements b {
public void methoda() {
System.out.println("methoda");
}

public void methodb() {
System.out.println("methodb");

}

}

class Interface4 {
public static void main(String[] args) {
c obj= new c();
obj.methoda();
obj.methodb();
}
}
```

SCREENSHOT WITH OUTPUT:-

Run Debug Stop Share Save { } Beautify

Interface4.java

```
interface a {
void methoda();
}

interface b extends a {
void methodb();
}

class c implements b {
public void methoda() {
System.out.println("methoda");
}

public void methodb() {
System.out.println("methodb");

}

}

class Interface4 {
public static void main(String[] args) {
c obj= new c();
obj.methoda();
obj.methodb();
}
}
```

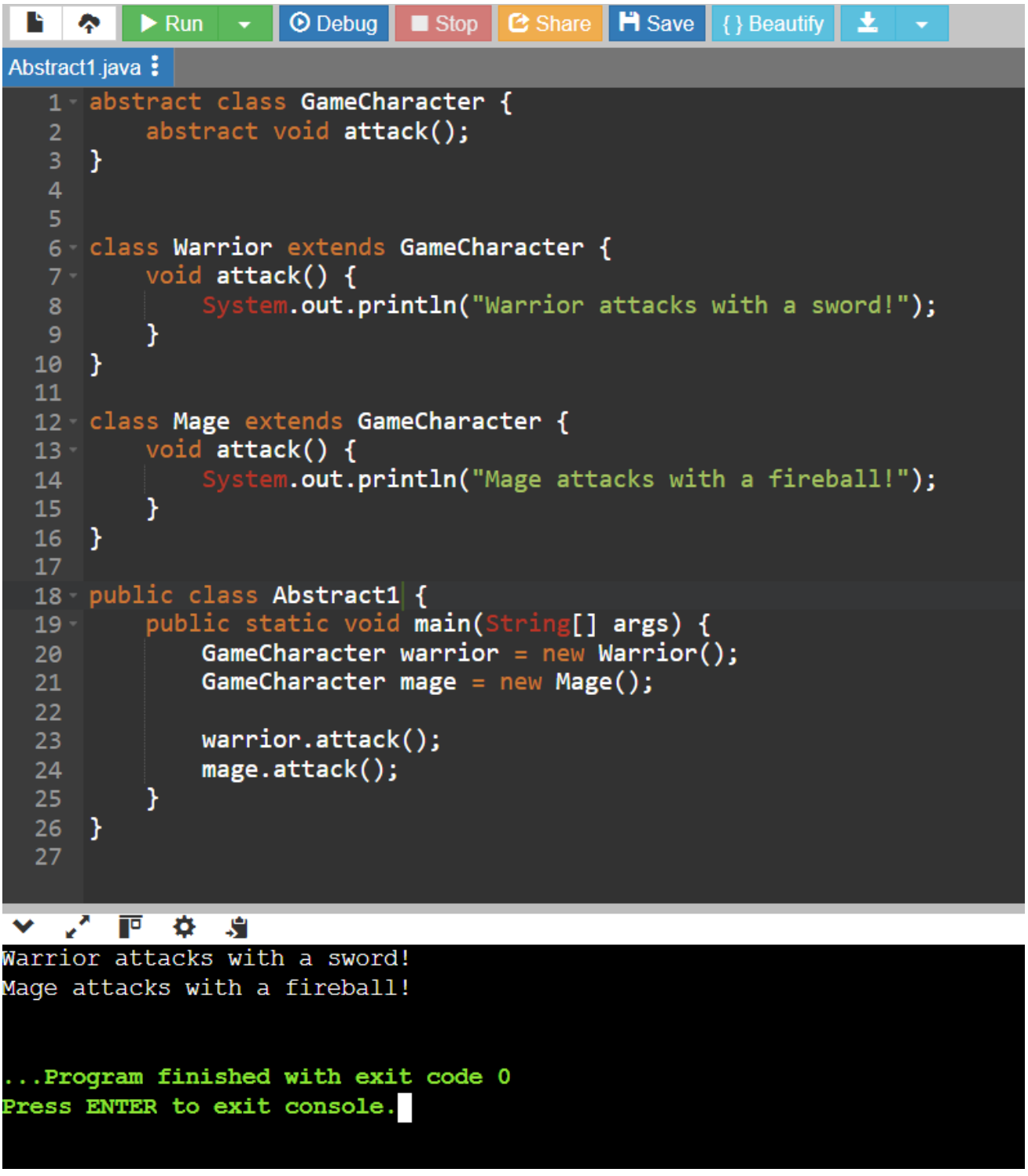
```
methoda
methodb


...Program finished with exit code 0
Press ENTER to exit console.
```

14.Abstract class Programes:-

a)

```
abstract class GameCharacter {
   abstract void attack();
}


class Warrior extends GameCharacter {
   void attack() {
      System.out.println("Warrior attacks with a sword!");
   }
}

class Mage extends GameCharacter {
   void attack() {
      System.out.println("Mage attacks with a fireball!");
   }
}

public class Abstract1 {
   public static void main(String[] args) {
      GameCharacter warrior = new Warrior();
      GameCharacter mage = new Mage();

      warrior.attack();
      mage.attack();
   }
}
```

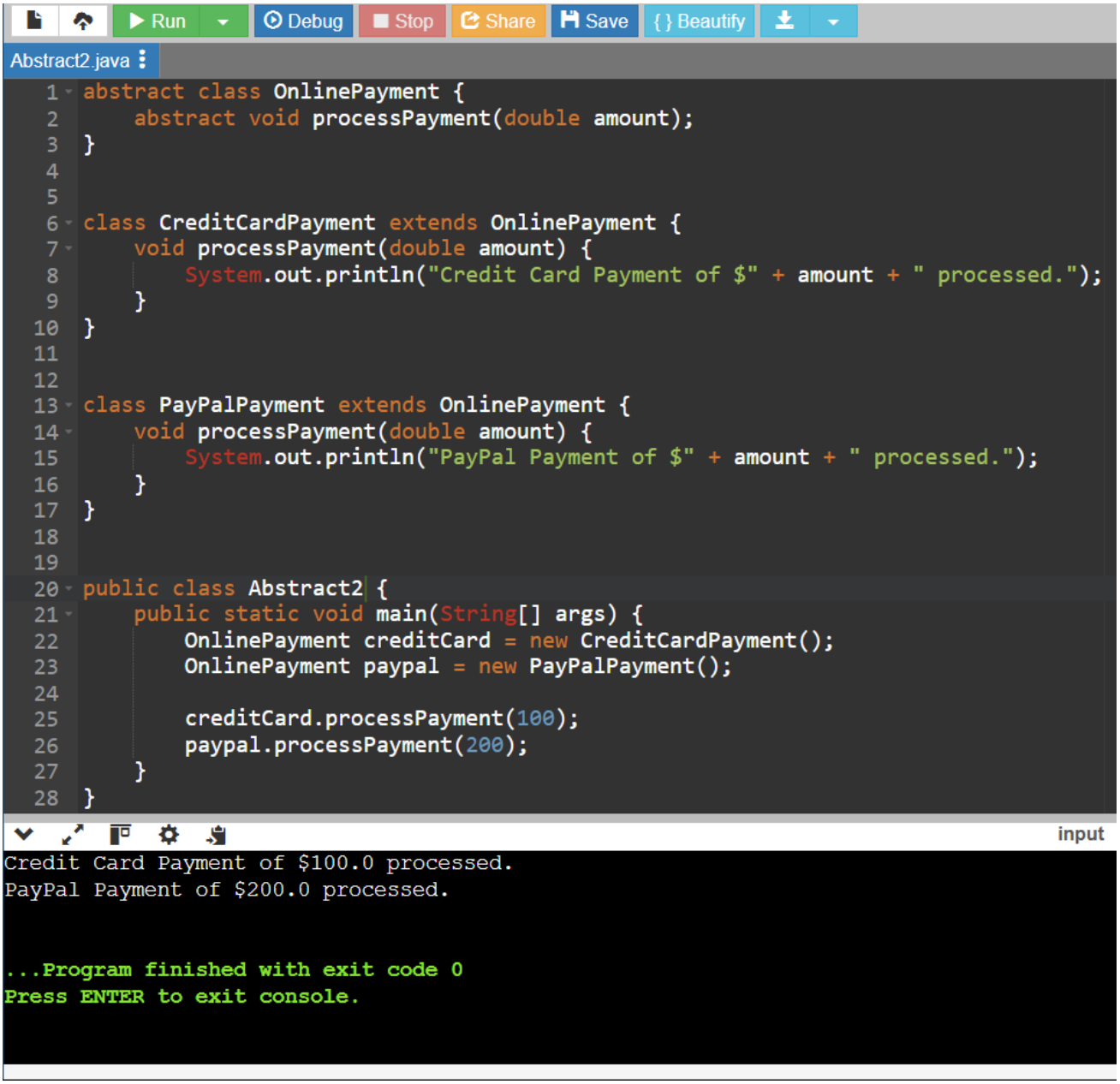
SCREENSHOT WITH OUTPUT:-



b)

```
abstract class OnlinePayment {
   abstract void processPayment(double amount);
}


class CreditCardPayment extends OnlinePayment {
   void processPayment(double amount) {
      System.out.println("Credit Card Payment of $" + amount + " processed.");
   }
}


class PayPalPayment extends OnlinePayment {
   void processPayment(double amount) {
      System.out.println("PayPal Payment of $" + amount + " processed.");
   }
}


public class Abstract2 {
   public static void main(String[] args) {
      OnlinePayment creditCard = new CreditCardPayment();
      OnlinePayment paypal = new PayPalPayment();

      creditCard.processPayment(100);
      paypal.processPayment(200);
   }
}
```

SCREENSHOT WITH OUTPUT:-



c)

```
abstract class BankAccount {
   String accountNumber;
   double balance;

   public BankAccount(String accountNumber, double balance) {
      this.accountNumber = accountNumber;
      this.balance = balance;
   }


   public void deposit(double amount) {
      balance += amount;
      System.out.println("Deposited " + amount + ". New balance: " + balance);
   }


   abstract void withdraw(double amount);
}


class SavingsAccount extends BankAccount {
   public SavingsAccount(String accountNumber, double balance) {
      super(accountNumber, balance);
   }


   void withdraw(double amount) {
      if (balance - amount >= 1000) {
         balance -= amount;
         System.out.println("Withdrawn " + amount + ". Remaining balance: " + balance);
      } else {
         System.out.println("Withdrawal denied! Minimum balance of 1000 must be maintained.");
      }
   }
}


class CurrentAccount extends BankAccount {
   public CurrentAccount(String accountNumber, double balance) {
      super(accountNumber, balance);
   }


   void withdraw(double amount) {
```

```
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawn " + amount + ". Remaining balance: " + balance);
        } else {
            System.out.println("Insufficient balance!");
        }
    }
}


public class Abstract3 {
    public static void main(String[] args) {
        BankAccount savings = new SavingsAccount("SA12345", 5000);
        BankAccount current = new CurrentAccount("CA67890", 10000);

        savings.deposit(2000);
        savings.withdraw(6000);
        savings.withdraw(2000);

        current.withdraw(5000);
        current.withdraw(6000);
    }
}
```

SCREENSHOT WITH OUTPUT:-

Run Debug Stop Share Save { } Beautify

Abstract3.java

```java
abstract class BankAccount {
    String accountNumber;
    double balance;

    public BankAccount(String accountNumber, double balance) {
        this.accountNumber = accountNumber;
        this.balance = balance;
    }


    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited " + amount + ". New balance: " + balance);
    }


    abstract void withdraw(double amount);
}


class SavingsAccount extends BankAccount {
    public SavingsAccount(String accountNumber, double balance) {
        super(accountNumber, balance);
    }


    void withdraw(double amount) {
        if (balance - amount >= 1000) {
            balance -= amount;
            System.out.println("Withdrawn " + amount + ". Remaining balance: " + balance);
        } else {
```

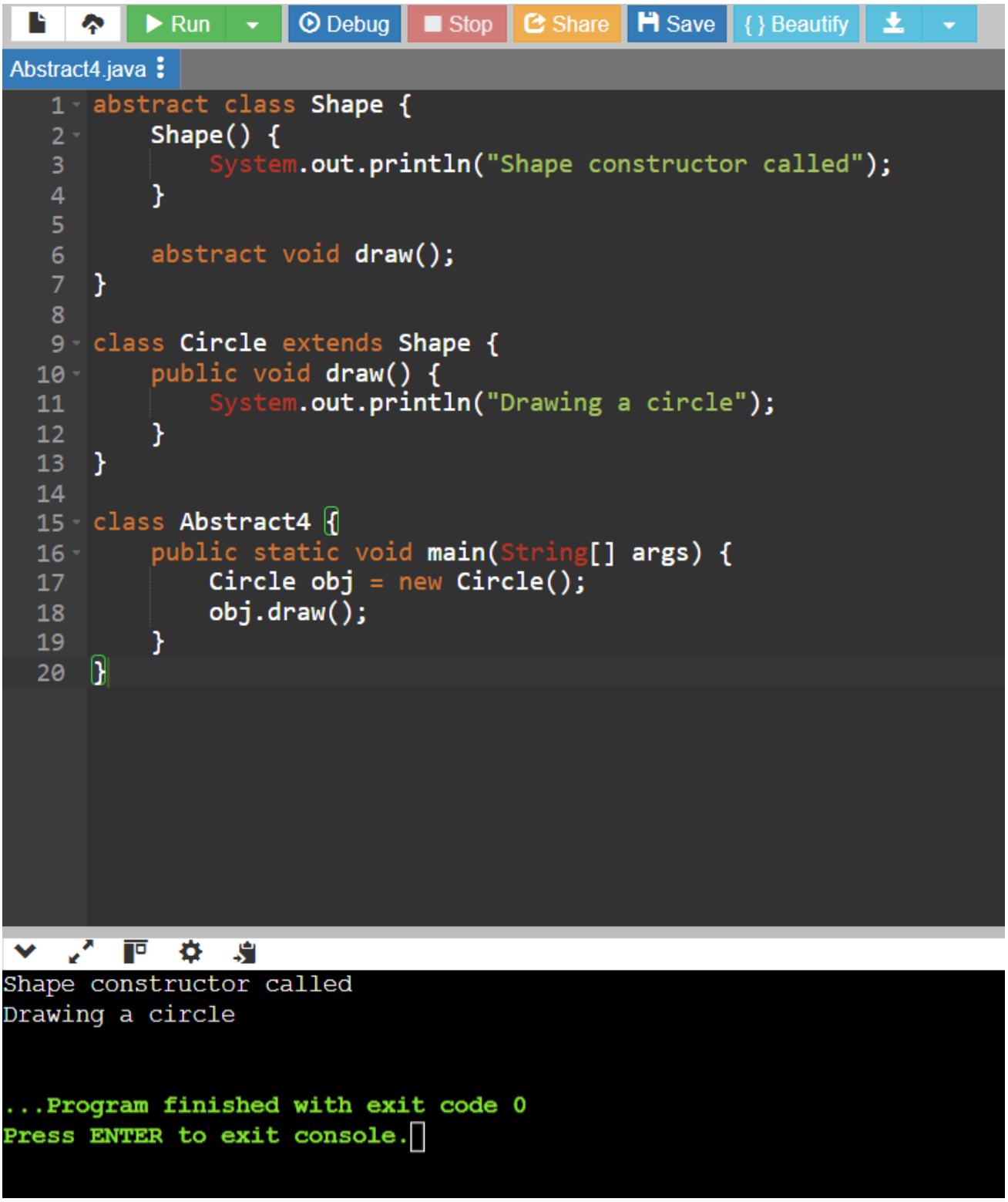
input

```
Deposited 2000.0. New balance: 7000.0
Withdrawn 6000.0. Remaining balance: 1000.0
Withdrawal denied! Minimum balance of 1000 must be maintained.
Withdrawn 5000.0. Remaining balance: 5000.0
Insufficient balance!
```

d)

```
abstract class Shape {
   Shape() {
      System.out.println("Shape constructor called");
   }

   abstract void draw();
}

class Circle extends Shape {
   public void draw() {
      System.out.println("Drawing a circle");
   }
}

class Abstract4 {
   public static void main(String[] args) {
      Circle obj = new Circle();
      obj.draw();
   }
}
```

SCREENSHOT WITH OUTPUT:-



ENCAPSULATION

15.Encapsulation Programes:-

a)

```
class Student {
   private String name;

   public void setName(String newName) {
      name = newName;
   }

   public String getName() {
      return name;
   }
}

class Encap1 {
   public static void main(String[] args) {
      Student s = new Student();
      s.setName("Manish");
      System.out.println("Student Name: " + s.getName());
   }
}
```

SCREENSHOT WITH OUTPUT:-

b)

```
import java.util.Scanner;

class students{
   private String name;
   private int roll;
   private String grade;

   public void setname(String name1){
      name=name1;
   }

   public void setgrade(String grade1){
      grade=grade1;
   }

   public void setroll(int roll1){
      roll=roll1;
   }



   public String getname(){
      return name;
   }

   public String getgrade(){
      return grade;
   }

   public int getroll(){
      return roll;
   }


}
public class Encap2 {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("name - ");
      String name1 = sc.nextLine();
      System.out.println("");
      System.out.print("grade - ");
      String grade1 = sc.nextLine();
      System.out.println("");
```

```
        System.out.print("roll - ");
        int roll1 = sc.nextInt();
        students obj = new students();
        obj.setname(name1);

        obj.setgrade(grade1);

        obj.setroll(roll1);

        System.out.println("name="+" "+obj.getname());
        System.out.println("roll="+" "+obj.getroll());
        System.out.println("grade="+" "+obj.getgrade());
    }

}
```

SCREENSHOT WITH OUTPUT:-

Run Debug Stop Share Save { } Beautify

Encap2.java

```java
import java.util.Scanner;

class students{
    private String name;
    private int roll;
    private String grade;

    public void setname(String name1){
        name=name1;
    }

    public void setgrade(String grade1){
        grade=grade1;
    }

    public void setroll(int roll1){
        roll=roll1;
    }



    public String getname(){
        return name;
    }

    public String getgrade(){
        return grade;
    }
```

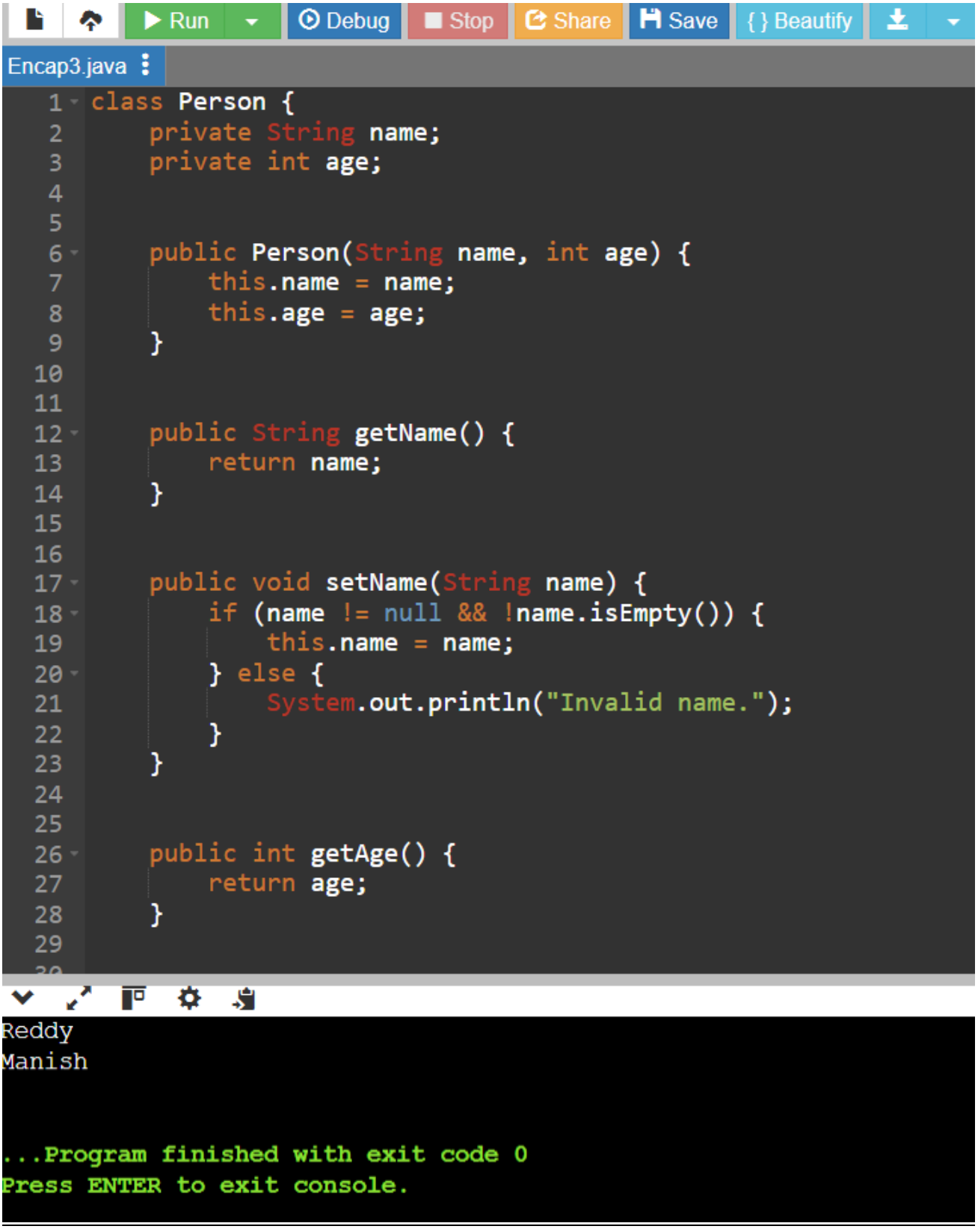
```
name - Manish

grade - 1st year

roll - 129
name= Manish
roll= 129
grade= 1st year
```

c)

```
class Person {
   private String name;
   private int age;


   public Person(String name, int age) {
      this.name = name;
      this.age = age;
   }


   public String getName() {
      return name;
   }


   public void setName(String name) {
      if (name != null && !name.isEmpty()) {
         this.name = name;
      } else {
         System.out.println("Invalid name.");
      }
   }


   public int getAge() {
      return age;
   }


   public void setAge(int age) {
      if (age > 0) {
         this.age = age;
      } else {
         System.out.println("Invalid age.");
      }
   }
}

public class Encap3 {
   public static void main(String[] args) {
      Person person = new Person("Reddy", 25);
```

```
        System.out.println(person.getName());
        person.setName("Manish");
        System.out.println(person.getName());
    }
}
```

SCREENSHOT WITH OUTPUT:-



d)

```
public class Encap4 {
    public static void main(String[] args) {
        Car car = new Car("Toyota", "Camry");
        car.setSpeed(120);
        System.out.println("Car: " + car.getBrand() + " " + car.getModel() + ", Speed: " +
car.getSpeed());
    }
}

class Car {
    private String brand;
    private String model;
    private int speed;

    public Car(String brand, String model) {
        this.brand = brand;
        this.model = model;
        this.speed = 0; // Default speed
    }

    public String getBrand() {
        return brand;
    }

    public String getModel() {
        return model;
    }

    public int getSpeed() {
        return speed;
    }

    public void setSpeed(int speed) {
        if (speed >= 0 && speed <= 200) {
            this.speed = speed;
        } else {
            System.out.println("Speed must be between 0 and 200.");
        }
    }
}
```

SCREENSHOT WITH OUTPUT:-

16.Packages Programes:-

a)

```
package mypackage;

public class UserDefined1 {
   public int add(int a, int b) {
      return a + b;
   }
   public int subtract(int a, int b) {
      return a - b;
   }
   public int multiply(int a, int b) {
      return a * b;
   }
   public double divide(int a, int b) {
      if (b != 0) {
         return (double) a / b;
      } else {
         System.out.println("Cannot divide by zero.");
         return 0;
      }
   }
   public static void main(String[] args) {
      UserDefined1 calc = new UserDefined1();
      int sum = calc.add(10, 5);
      int difference = calc.subtract(10, 5);
      int product = calc.multiply(10, 5);
      double quotient = calc.divide(10, 5);
      double invalidQuotient = calc.divide(10, 0);
      System.out.println("Addition: 10 + 5 = " + sum);
      System.out.println("Subtraction: 10 - 5 = " + difference);
      System.out.println("Multiplication: 10 * 5 = " + product);
      System.out.println("Division: 10 / 5 = " + quotient);
      System.out.println("Division by zero: 10 / 0 = " + invalidQuotient);
   }
}
```

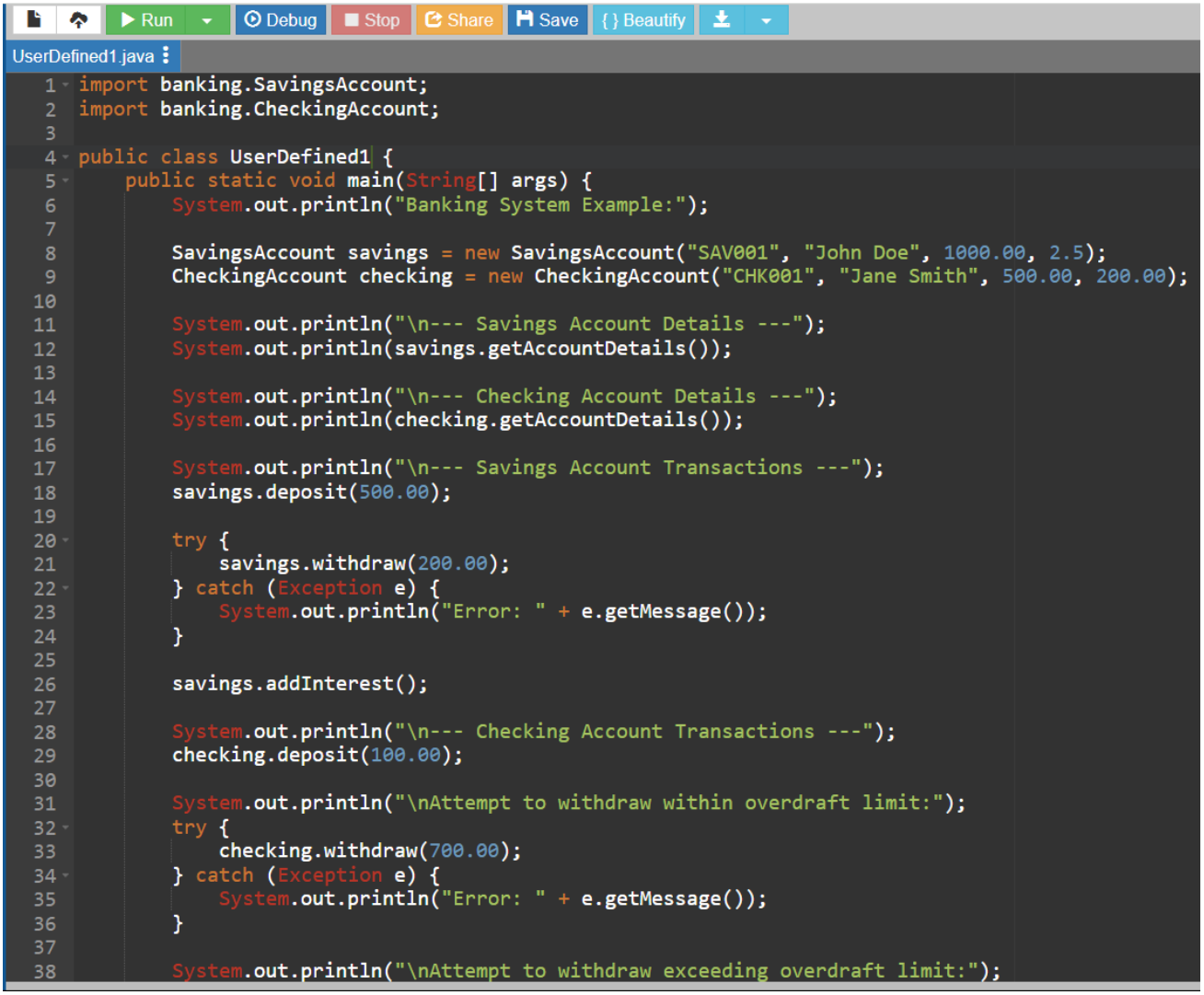
SCREENSHOT WITH OUTPUT:-

b)

```
import banking.SavingsAccount;
import banking.CheckingAccount;

public class UserDefined2 {
   public static void main(String[] args) {
      System.out.println("Banking System Example:");

      SavingsAccount savings = new SavingsAccount("SAV001", "John Doe", 1000.00, 2.5);
      CheckingAccount checking = new CheckingAccount("CHK001", "Jane Smith", 500.00, 200.00);

      System.out.println("\n--- Savings Account Details ---");
      System.out.println(savings.getAccountDetails());

      System.out.println("\n--- Checking Account Details ---");
      System.out.println(checking.getAccountDetails());

      System.out.println("\n--- Savings Account Transactions ---");
      savings.deposit(500.00);

      try {
         savings.withdraw(200.00);
      } catch (Exception e) {
         System.out.println("Error: " + e.getMessage());
      }

      savings.addInterest();

      System.out.println("\n--- Checking Account Transactions ---");
      checking.deposit(100.00);

      System.out.println("\nAttempt to withdraw within overdraft limit:");
      try {
         checking.withdraw(700.00);
      } catch (Exception e) {
         System.out.println("Error: " + e.getMessage());
      }

      System.out.println("\nAttempt to withdraw exceeding overdraft limit:");
      try {
         checking.withdraw(1000.00);
      } catch (Exception e) {
         System.out.println("Error: " + e.getMessage());
      }
```

```
        System.out.println("\nException Handling Examples:");
        try {
            savings.withdraw(-100.00);
        } catch (IllegalArgumentException e) {
            System.out.println("Caught exception: " + e.getMessage());
        }
    }
}
```

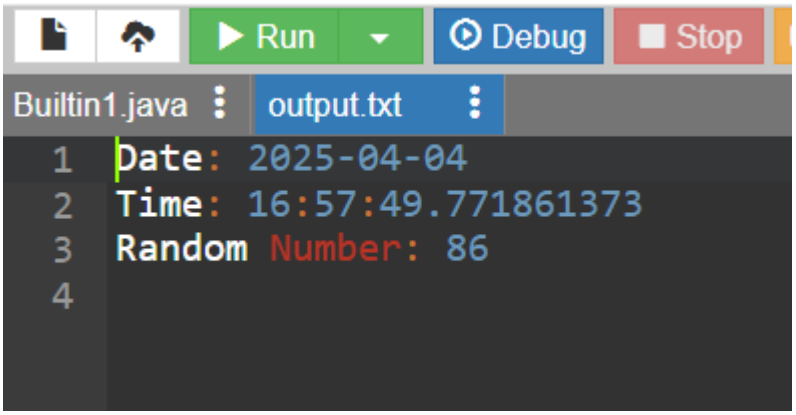
SCREENSHOT WITH OUTPUT:-



c)

```
import java.time.LocalDate;
import java.time.LocalTime;
import java.util.Random;
import java.util.Date;
import java.io.FileWriter;
import java.io.IOException;

public class Builtin1 {
   public static void main(String[] args) {
      LocalDate today = LocalDate.now();
      LocalTime now = LocalTime.now();
      System.out.println("Today's Date: " + today);
      System.out.println("Current Time: " + now);

      Random random = new Random();
      int randNumber = random.nextInt(100);
      System.out.println("Random Number: " + randNumber);

      // File
      try {
         FileWriter writer = new FileWriter("output.txt");
         writer.write("Date: " + today + "\n");
         writer.write("Time: " + now + "\n");
         writer.write("Random Number: " + randNumber + "\n");
         writer.close();
         System.out.println("Data successfully written to file.");
      } catch (IOException e) {
         e.printStackTrace();
      }
   }
}
```

SCREENSHOT WITH OUTPUT:-



Run Debug Stop Share Save { } Beautify

Builtin1.java output.txt

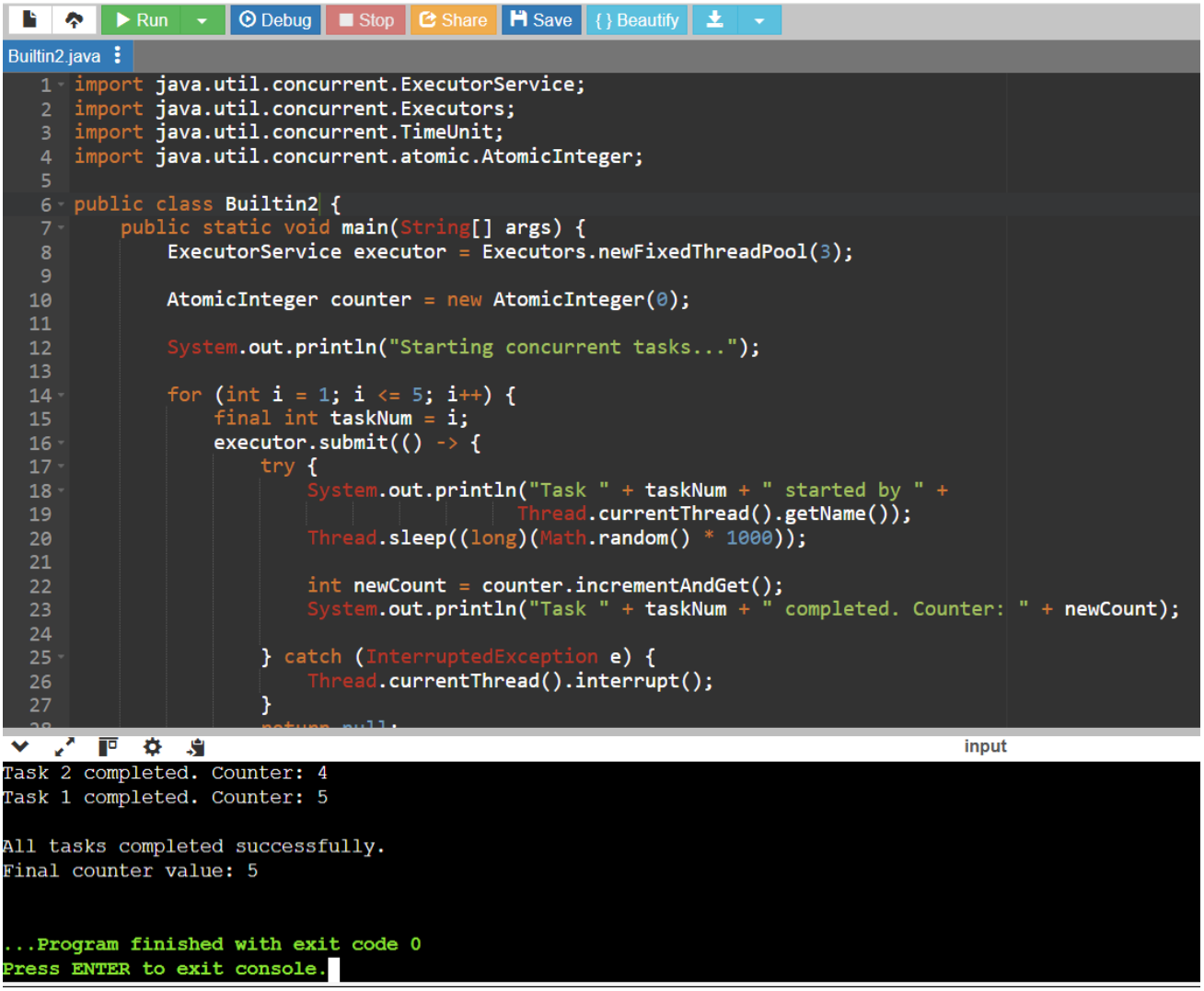
```java
import java.time.LocalDate;
import java.time.LocalTime;
import java.util.Random;
import java.util.Date;
import java.io.FileWriter;
import java.io.IOException;

public class Builtin1 {
    public static void main(String[] args) {
        LocalDate today = LocalDate.now();
        LocalTime now = LocalTime.now();
        System.out.println("Today's Date: " + today);
        System.out.println("Current Time: " + now);

        Random random = new Random();
        int randNumber = random.nextInt(100);
        System.out.println("Random Number: " + randNumber);

        // File
        try {
            FileWriter writer = new FileWriter("output.txt");
            writer.write("Date: " + today + "\n");
            writer.write("Time: " + now + "\n");
            writer.write("Random Number: " + randNumber + "\n");
            writer.close();
            System.out.println("Data successfully written to file.");
        } catch (IOException e) {
```

```
Today's Date: 2025-04-04
Current Time: 16:57:49.771861373
Random Number: 86
Data successfully written to file.


...Program finished with exit code 0
Press ENTER to exit console.
```

d)

```
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.TimeUnit;
import java.util.concurrent.atomic.AtomicInteger;

public class Builtin2 {
   public static void main(String[] args) {
      ExecutorService executor = Executors.newFixedThreadPool(3);

      AtomicInteger counter = new AtomicInteger(0);

      System.out.println("Starting concurrent tasks...");

      for (int i = 1; i <= 5; i++) {
         final int taskNum = i;
         executor.submit(() -> {
            try {
               System.out.println("Task " + taskNum + " started by " +
                        Thread.currentThread().getName());
               Thread.sleep((long)(Math.random() * 1000));

               int newCount = counter.incrementAndGet();
               System.out.println("Task " + taskNum + " completed. Counter: " + newCount);

            } catch (InterruptedException e) {
               Thread.currentThread().interrupt();
            }
            return null;
         });
      }

      executor.shutdown();
      try {
         if (!executor.awaitTermination(5, TimeUnit.SECONDS)) {
            System.out.println("Not all tasks completed in time.");
            executor.shutdownNow();
         } else {
            System.out.println("\nAll tasks completed successfully.");
            System.out.println("Final counter value: " + counter.get());
         }
      } catch (InterruptedException e) {
         executor.shutdownNow();
         Thread.currentThread().interrupt();
      }
   }
}
```

SCREENSHOT WITH OUTPUT:-

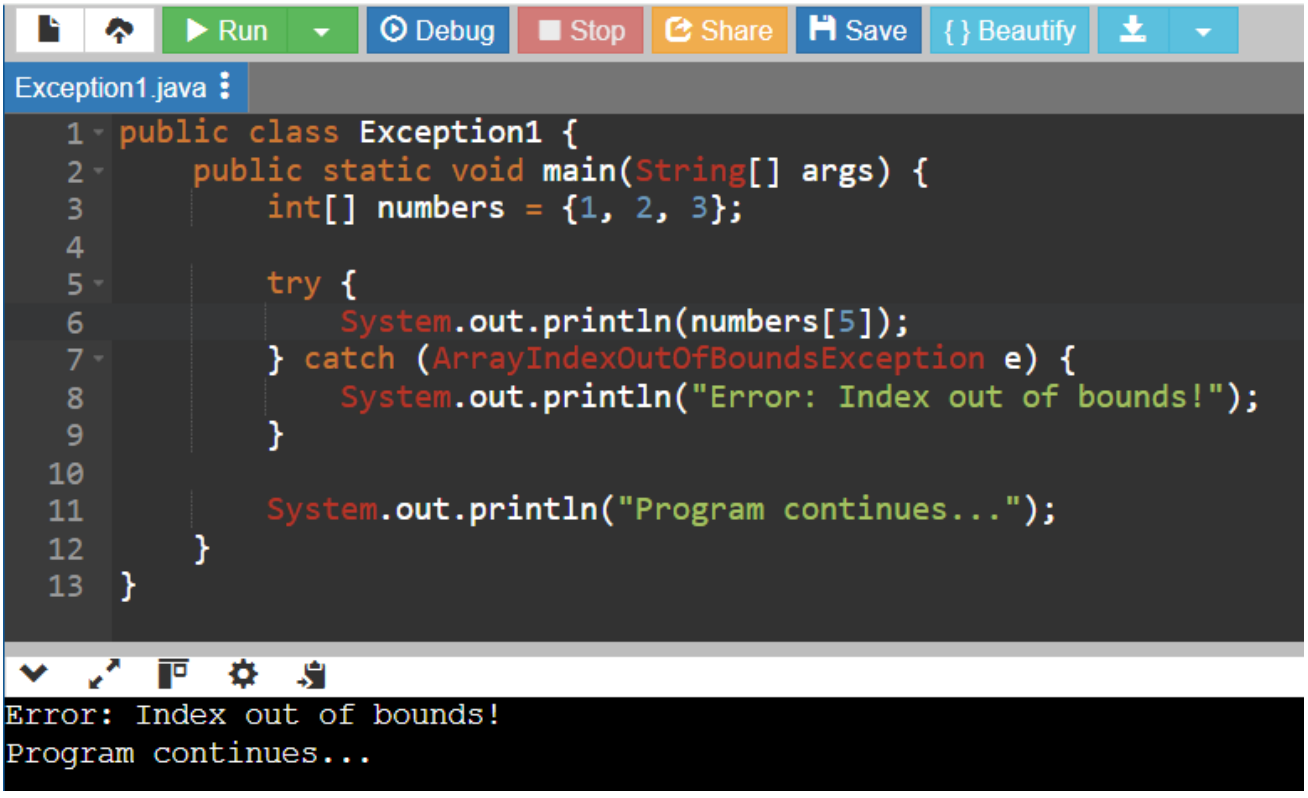


17.Exception Handiling Programes:-

a)

```
public class Exception1 {
   public static void main(String[] args) {
      int[] numbers = {1, 2, 3};

      try {
         System.out.println(numbers[5]);
      } catch (ArrayIndexOutOfBoundsException e) {
         System.out.println("Error: Index out of bounds!");
      }

      System.out.println("Program continues...");
   }
}
```

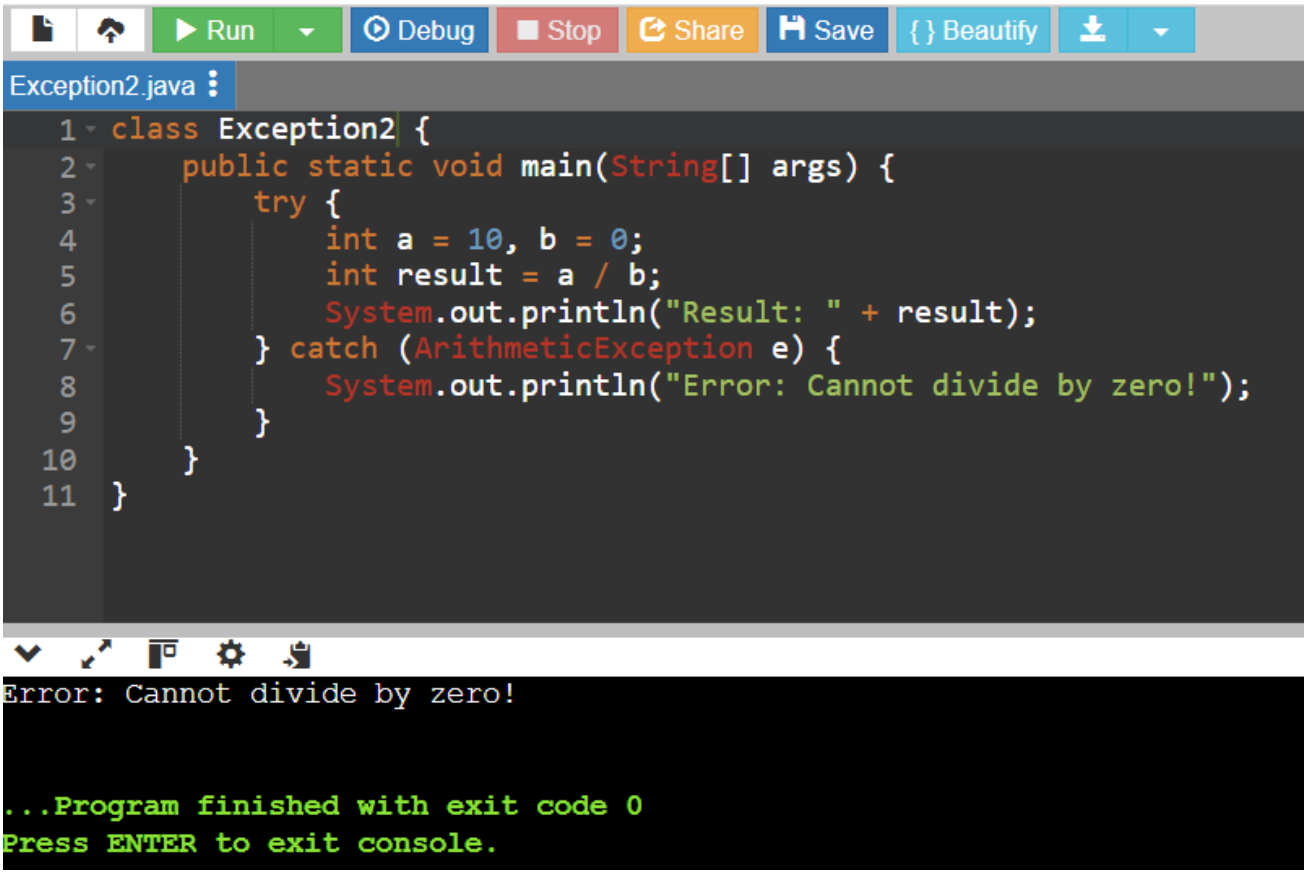
SCREENSHOT WITH OUTPUT:-



b)

```
class Exception2 {
   public static void main(String[] args) {
      try {
         int a = 10, b = 0;
         int result = a / b;
         System.out.println("Result: " + result);
      } catch (ArithmeticException e) {
         System.out.println("Error: Cannot divide by zero!");
      }
   }
}
```

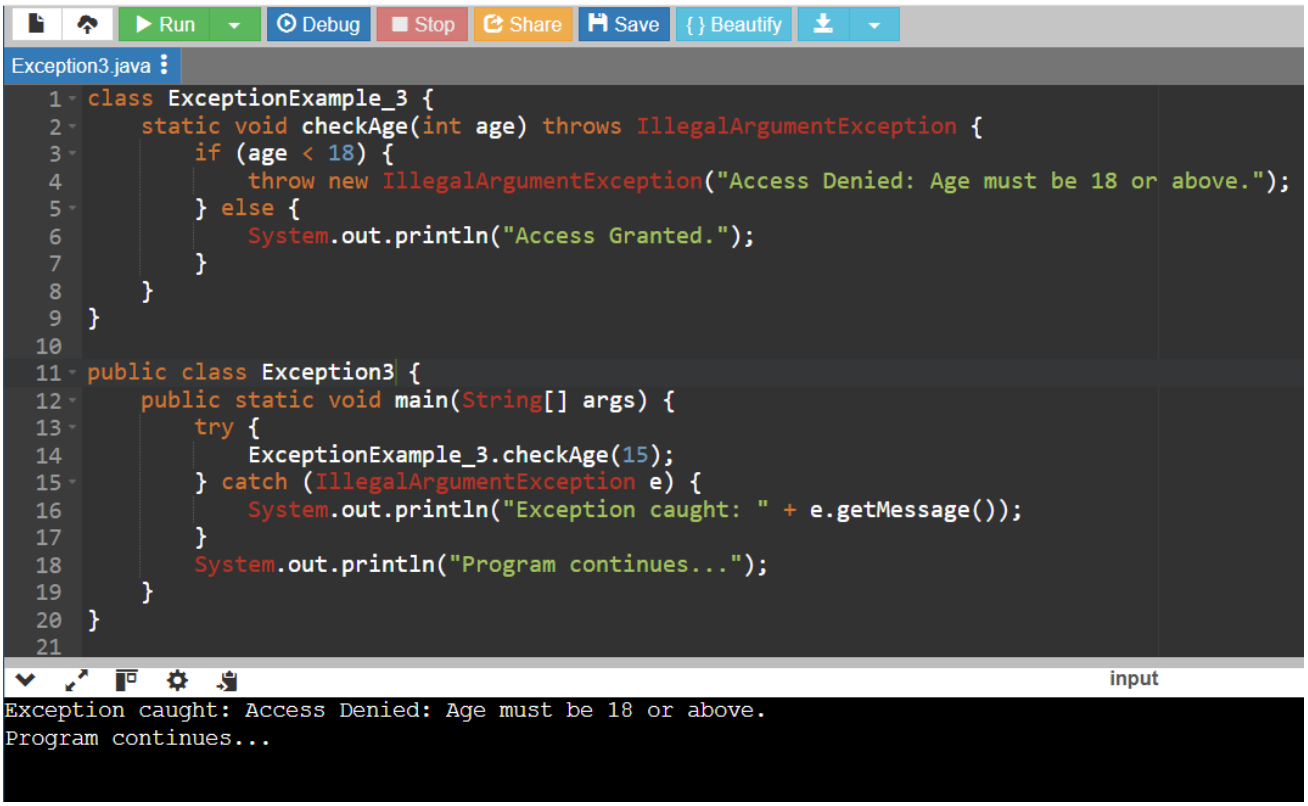
SCREENSHOT WITH OUTPUT:-



c)

```
class ExceptionExample_3 {
   static void checkAge(int age) throws IllegalArgumentException {
      if (age < 18) {
         throw new IllegalArgumentException("Access Denied: Age must be 18 or above.");
      } else {
         System.out.println("Access Granted.");
      }
   }
}

public class Exception3 {
   public static void main(String[] args) {
      try {
         ExceptionExample_3.checkAge(15);
      } catch (IllegalArgumentException e) {
         System.out.println("Exception caught: " + e.getMessage());
      }
      System.out.println("Program continues...");
   }
}
```

SCREENSHOT WITH OUTPUT:-



d)

```
import java.util.*;
public class Exception4 {
   public static void main(String[] args) {
      detal n= new detal();
      n.mm();
   }
}
class detal{
   void mm(){
      try{
      Scanner inp= new Scanner(System.in);
      System.out.println("enter your age");
      int age=inp.nextInt();
      if (age<1){
         throw new Exception("age should be more than 1");
      }
      }
      catch (Exception e){
         System.out.println(e);
      }
      finally {
         System.out.println("the programm has ended sucessfully");
      }
   }
}
```

SCREENSHOT WITH OUTPUT:-

Run Debug Stop Share Save { } Beautify

Exception4.java

```java
import java.util.*;
public class Exception4 {
    public static void main(String[] args) {
        detal n= new detal();
        n.mm();
    }
}
class detal{
    void mm(){
        try{
        Scanner inp= new Scanner(System.in);
        System.out.println("enter your age");
        int age=inp.nextInt();
        if (age<1){
            throw new Exception("age should be more than 1");
        }
        }
        catch (Exception e){
            System.out.println(e);
        }
        finally {
            System.out.println("the programm has ended sucessfully");
        }
    }
}
```

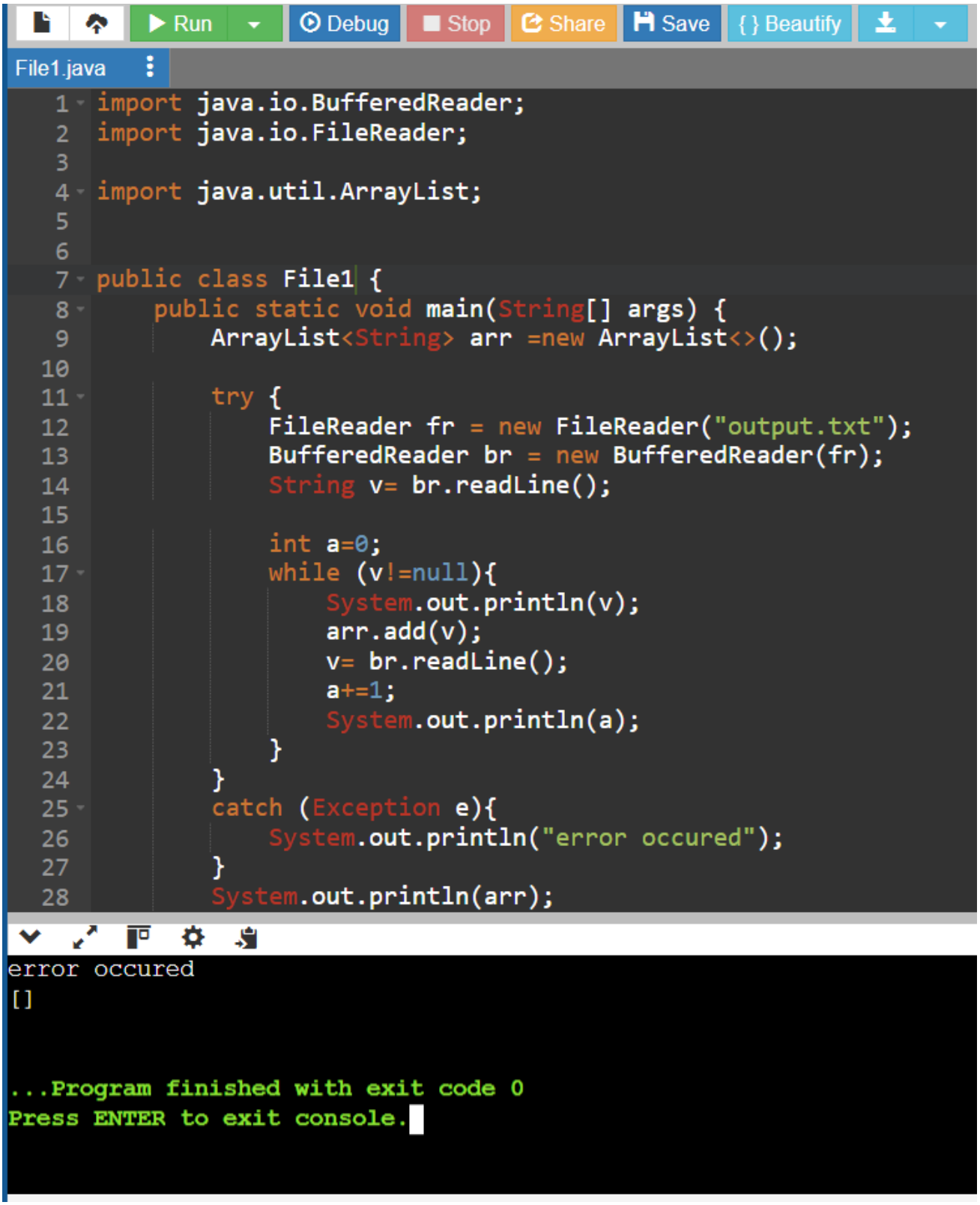
```
enter your age
18
the programm has ended sucessfully


...Program finished with exit code 0
Press ENTER to exit console.
```

18.File Handiling Programes:-

a)

```
import java.io.BufferedReader;
import java.io.FileReader;

import java.util.ArrayList;


public class File1 {
   public static void main(String[] args) {
      ArrayList<String> arr =new ArrayList<>();

      try {
         FileReader fr = new FileReader("output.txt");
         BufferedReader br = new BufferedReader(fr);
         String v= br.readLine();

         int a=0;
         while (v!=null){
            System.out.println(v);
            arr.add(v);
            v= br.readLine();
            a+=1;
            System.out.println(a);
         }
      }
      catch (Exception e){
         System.out.println("error occured");
      }
      System.out.println(arr);

   }
}
```

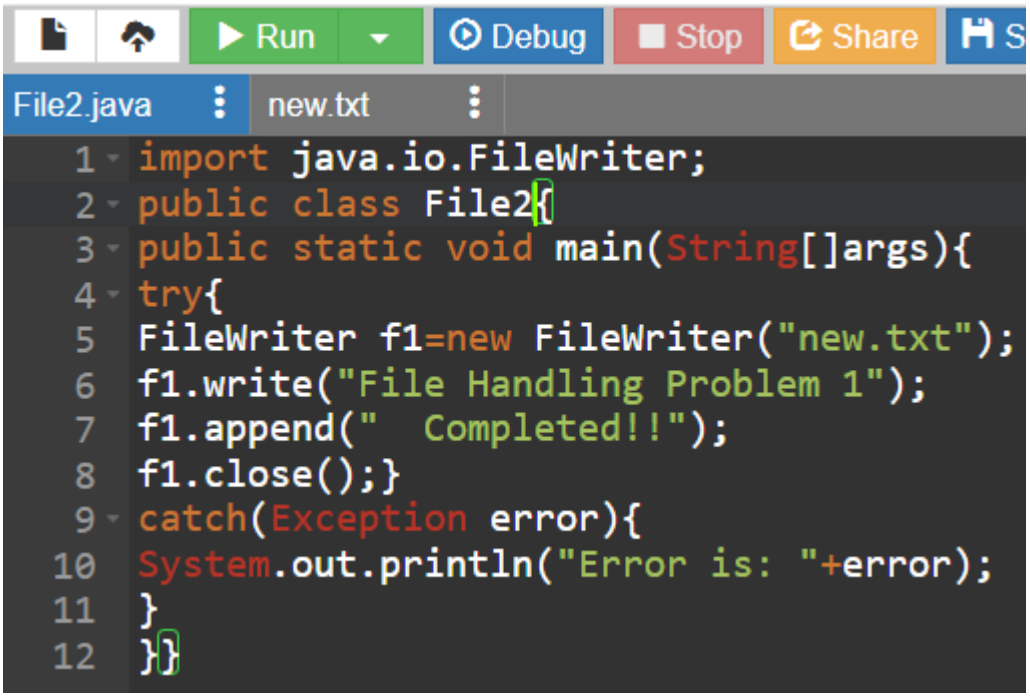
SCREENSHOT WITH OUTPUT:-

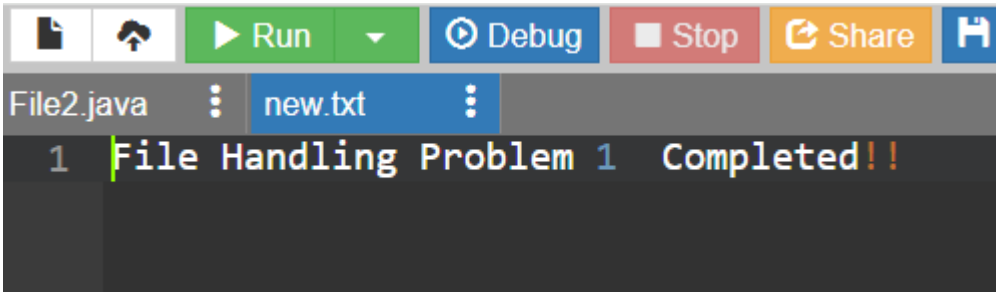


b)

```
import java.io.FileWriter;
public class File2{
public static void main(String[]args){
try{
FileWriter f1=new FileWriter("new.txt");
f1.write("File Handling Problem 1");
f1.append("  Completed!!");
f1.close();}
catch(Exception error){
System.out.println("Error is: "+error);
}
}}
```

SCREENSHOT WITH OUTPUT:-

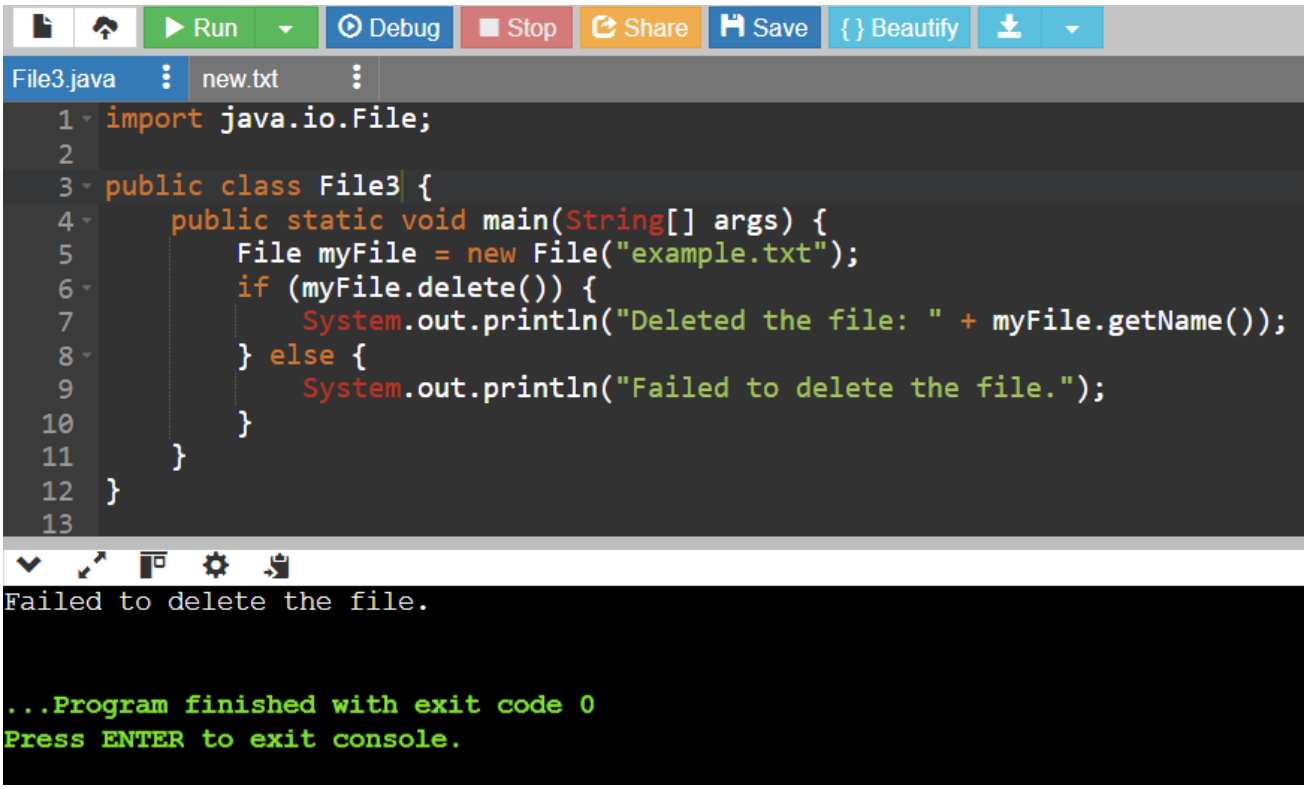




c)

```
import java.io.File;

public class File3 {
   public static void main(String[] args) {
      File myFile = new File("example.txt");
      if (myFile.delete()) {
         System.out.println("Deleted the file: " + myFile.getName());
      } else {
         System.out.println("Failed to delete the file.");
      }
   }
}
```

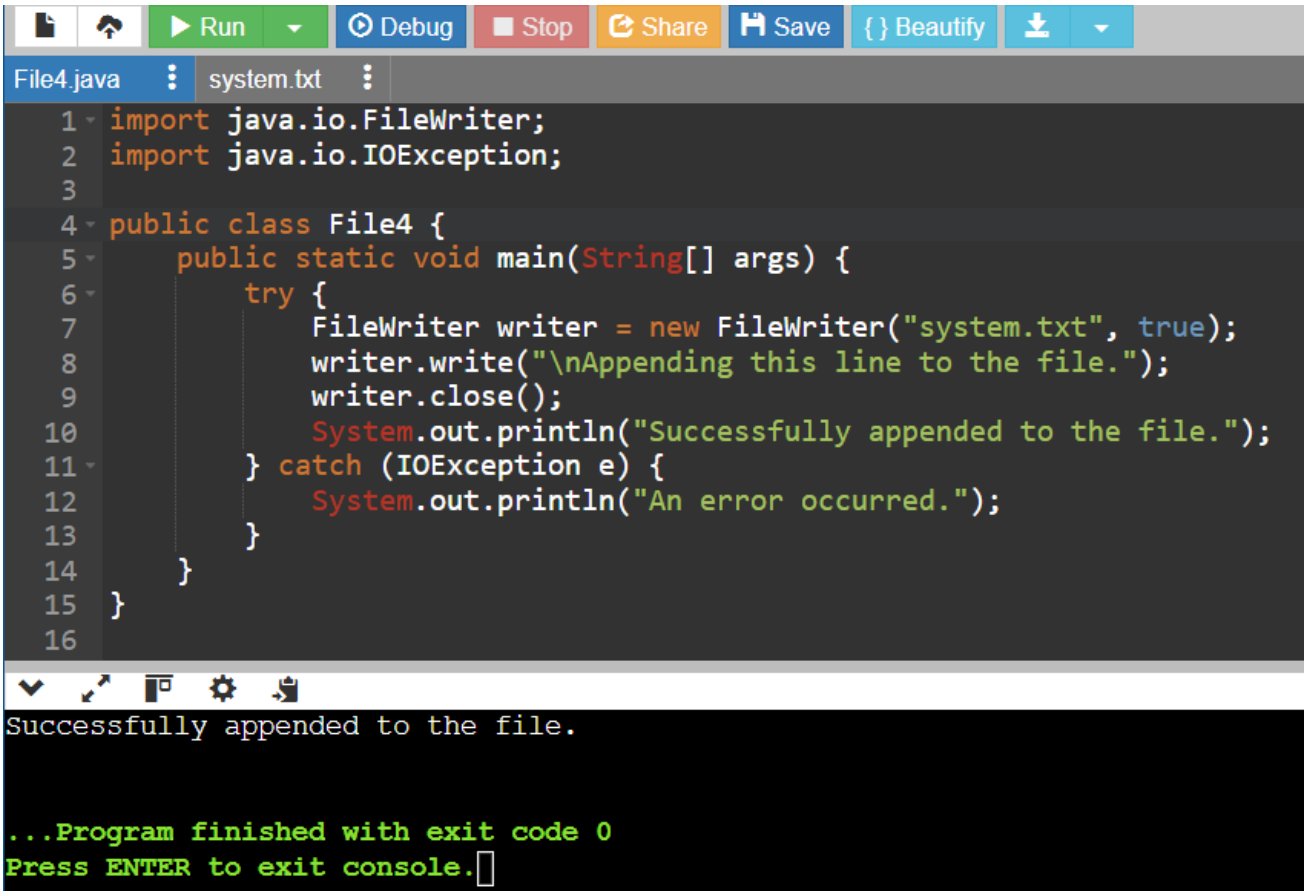
SCREENSHOT WITH OUTPUT:-

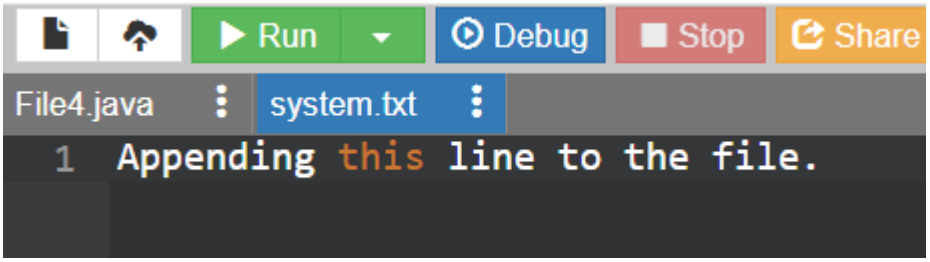


d)

```
import java.io.FileWriter;
import java.io.IOException;

public class File4 {
   public static void main(String[] args) {
      try {
         FileWriter writer = new FileWriter("system.txt", true);
         writer.write("\nAppending this line to the file.");
         writer.close();
         System.out.println("Successfully appended to the file.");
      } catch (IOException e) {
         System.out.println("An error occurred.");
      }
   }
}
```

SCREENSHOT WITH OUTPUT:-





Name:-Musali Reddy Manish Reddy
Registration no:-CH.SC.U4CSE24129