23CSE111

**OBJECT ORIENTED PROGRAMMING**

**DOCUMENT**

****

## Department of computer science Engineering

## Amrita School of Engineering

**Amrita Vishwa Vidyapeetham, Amaravati Campus**

### Name: Bhuvana Harshithaa Vudumula

**Verified By : Roll No: CSEA-24027**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. No | Programs | Date | Page No | Signature |
| WEEK 1 |  | 27-01-2025 |  |  |
| 1 | Write the steps to download and install Java. |  |  |  |
| 2 | Write a java program to print the  message “Welcome to java programming”. |  |  |  |
| 3 | Write a java program that prints name, roll number and section of a student. |  |  |  |
| WEEK 2 |  | 3-02-2025 |  |  |
| 1 | Write a java program to calculate the area of a rectangle. |  |  |  |
| 2a) | Write a program to convert temperature from Celsius to Fahrenheit |  |  |  |
| b) | Write a program to convert  temperature from Fahrenheit to Celsius. |  |  |  |
| 3 | Write a program to calculate the simple interest |  |  |  |
| 4 | Write a program to find the largest of three numbers using ternary operator. |  |  |  |
| 5 | Write a program to find the factorial of a number |  |  |  |
| WEEK 3 |  | 11-02-2025 |  |  |
| 1 | Creating a car class with the given instructions |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Creating a BankAccount class with the given instructions |  |  |  |
| WEEK 4 |  | 02-03-2025 |  |  |
| 1 | Write a java program with class named “Book” with given  instructions. |  |  |  |
| 2 | To create a java program with class named Myclass with given  instructions. |  |  |  |
| WEEK 5 |  | 09-03-2025 |  |  |
| 1 | Create a calc using the operations including add, sub, mul, div using multilevel inheritance and display the desired output. |  |  |  |
| 2 | Creating a Rental System. |  |  |  |
| WEEK 6 |  | 16-03-2025 |  |  |
| 1 | Write a java program to create a Vehicle class with displayInfo()  method , overridden in Car subclass  to provide info about carcompany , model , price ,seating and petrol. |  |  |  |
| 2 | An automated admission system that verifies student eligibility for UG and PG with different criteria.   1. UG requires minimum of 60% 2. PG requires minimum of 70% |  |  |  |
| 3 | Create a calculator class with overloaded methods to perform additions   1. add two integers |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1. add two double values 2. add three integers |  |  |  |
| 4 | Create a shape class with method calculateArea() that is overloaded for different shapes (eg: square, rectangle).Then create a subclass  Circle that overrides calculateArea()  method for Circle. |  |  |  |
| WEEK 7 |  | 14-04-2025 |  |  |
| 1 | Write a java program to create an abstract class Animal with abstract method sound and create subclasses Lion and Tiger that  implements the method. |  |  |  |
| 2 | Write a java program to create an abstract class shape3D with  abstract methods to calculate volume and surfacearea and create subclasses for sphere and cube that  implements these methods. |  |  |  |
| 3 | Create an abstract class PatternPrint with an abstract method printing to print the pattern and a concrete method to display the pattern .  Implement the patterns   1. Star Pattern - prints a right angled triangle of stars 2. Number Pattern – prints a right angled triangle of increasing   numbers. |  |  |  |

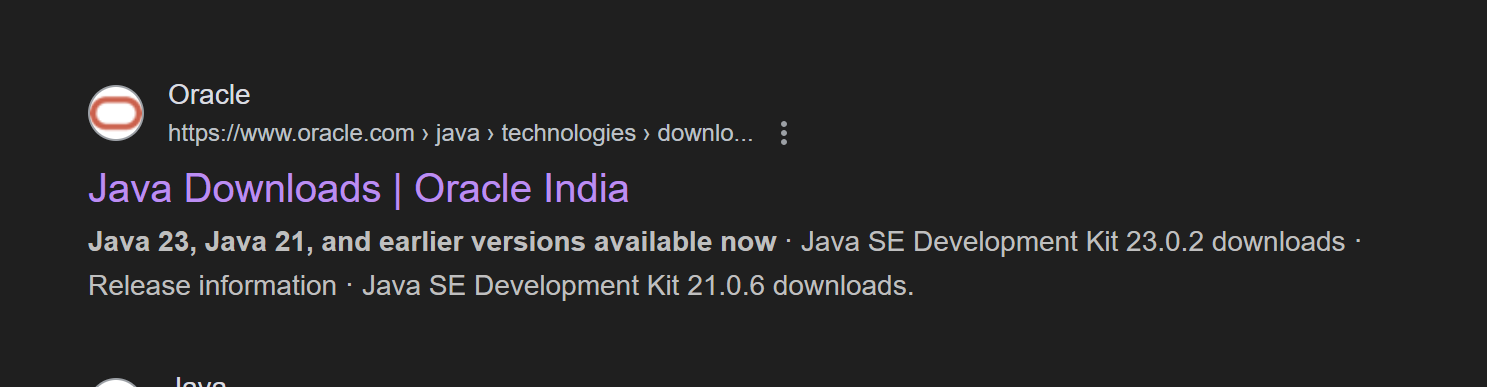
|  |  |  |
| --- | --- | --- |
|  |  |  |

**WEEK - 1**

1. **Write the steps to download and install Java.**

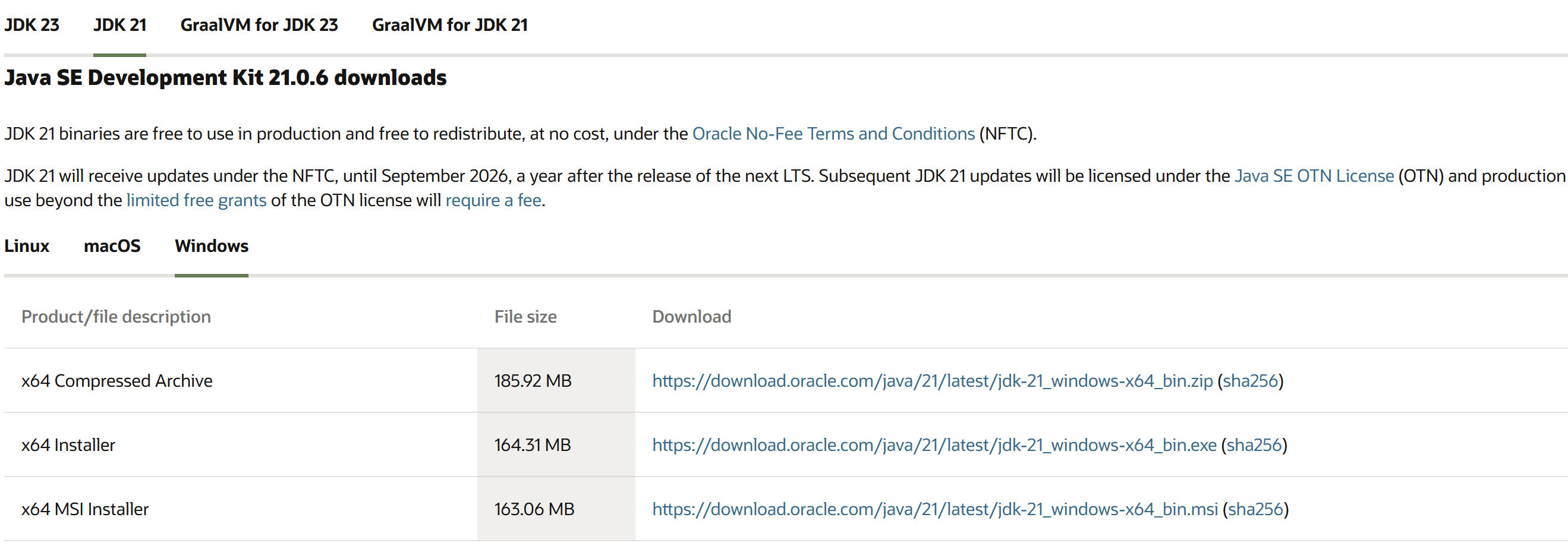
**Aim : Download and Install Java Software**

**Step – 1 : Visit any web browser and search for java download. Select the official Oracle website.**

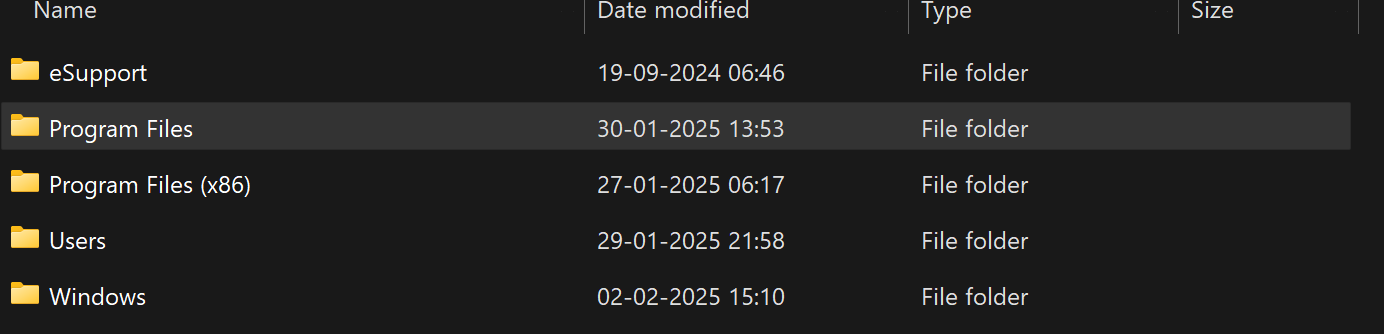


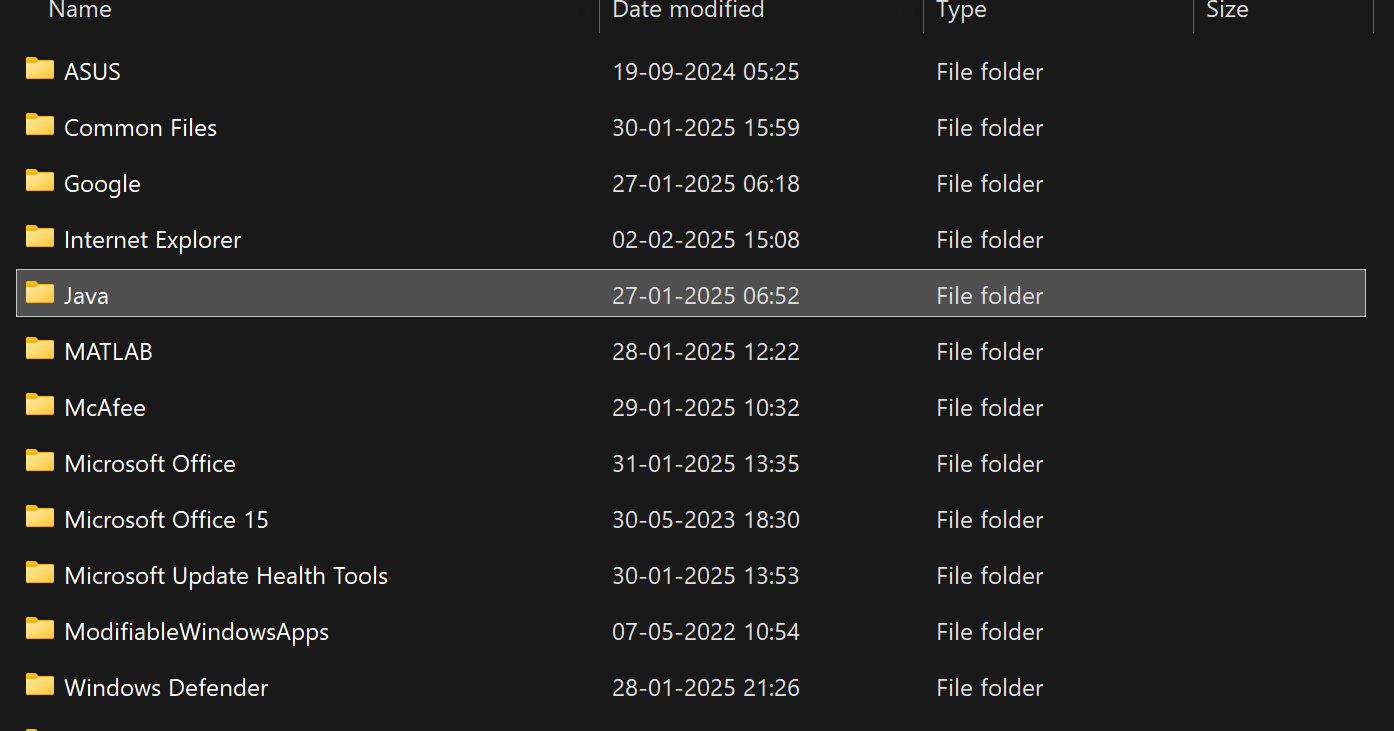
**Step – 2 : Open Oracle website and select the LTS “JDK 21 “ for Windows and select “X64 Installer” and**

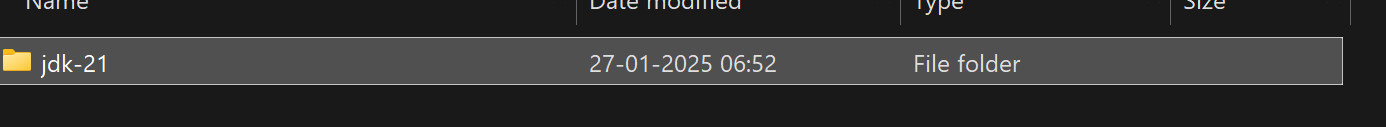
**download it.**



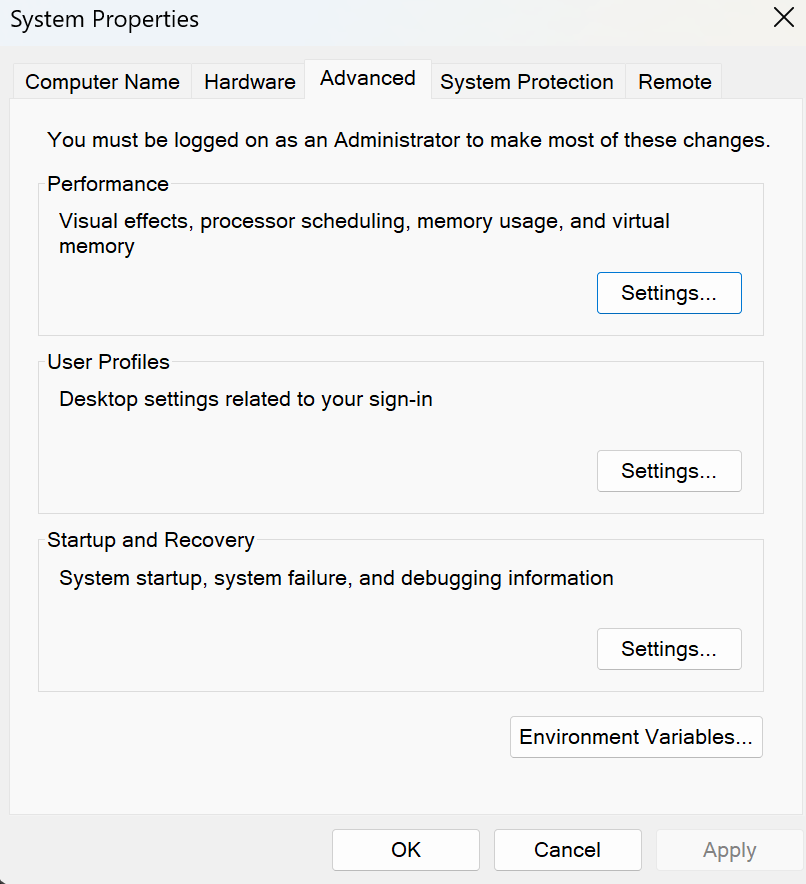
**Step – 3 : After downloading open “C-drive” on your pc and select “Program Files”, open “JDK 21”**

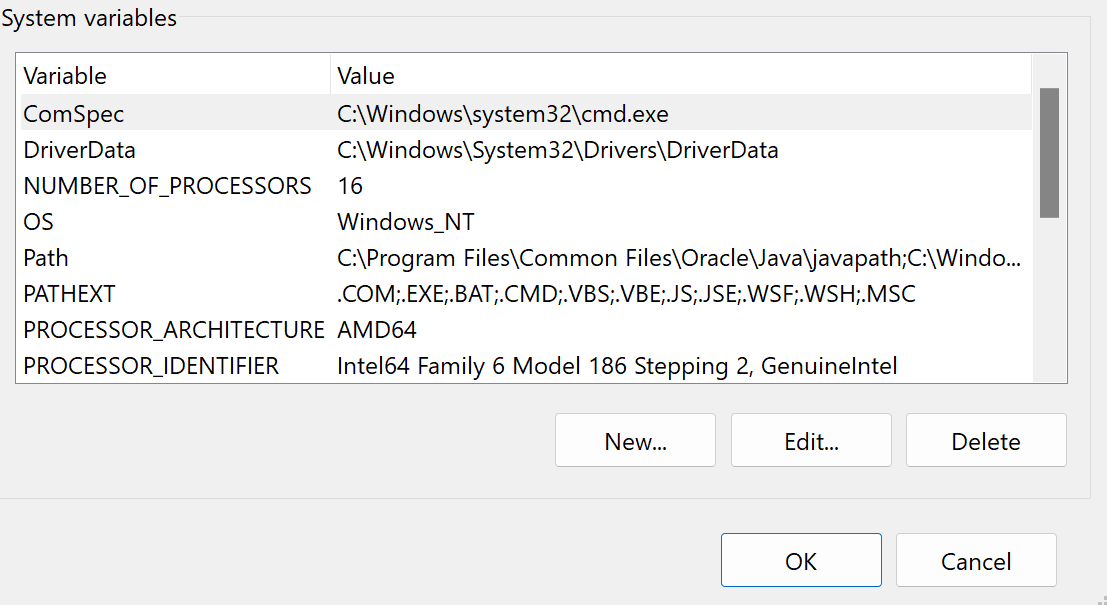




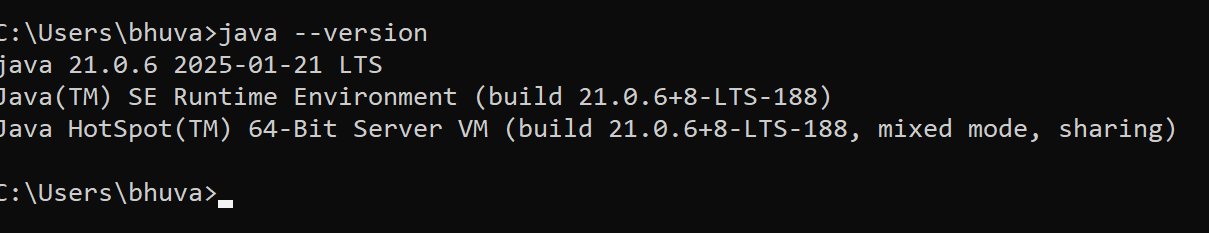


**Step – 4 : Open environmental variables and add a new file with path.**





**Step – 5 : Verify java version in command window**



1. **Write a java program to print the message “Welcome to java programming”.**

**Code:**

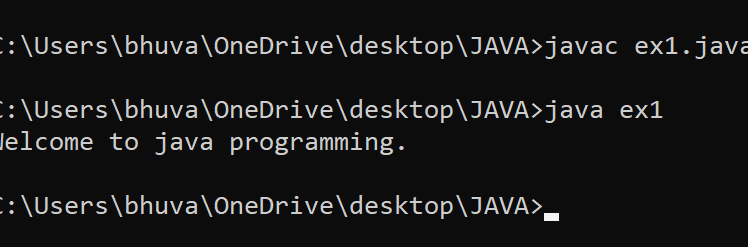
**class ex1 {**

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

**System.out.println("Welcome to java programming.");**

**}**

**}**



**Error :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **;** | **; is expected at end** |
| **2** | **S** | **Capital S is expected for String and System.** |

1. **Write a java program to print the student information**

**Code :**

**class ex2{**

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

**System.out.println("Student Information:");**

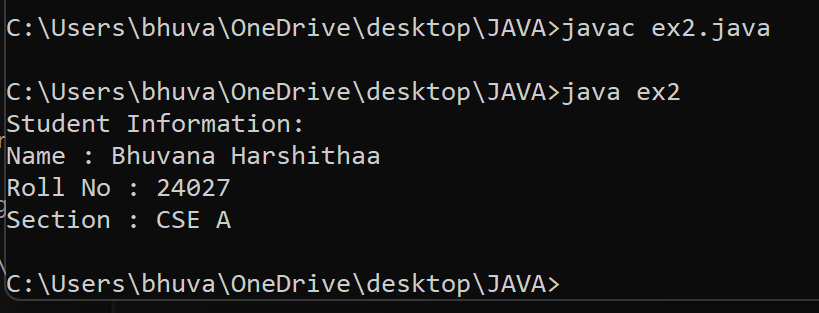
**System.out.println(“Name : Bhuvana Harshithaa”);**

**System.out.println(“Roll No : 24027”)**

**System.out.println(“Section : CSE A”)**

**}**

**}**



**ERRORS :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **;** | **; is expected at end** |
| **2** | **S** | **Capital S is expected for String** |

**WEEK – 2**

1. **Write a java program to clalculate area of rectangle.**

**Code : import java.util.Scanner;**

**public class arear{**

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

**Scanner input = new Scanner(System.in);**

**System.out.print("Enter a value : ");**

**int b = input.nextInt();**

**System.out.print("Enter a value : ");**

**int l = input.nextInt();**

**int area = b\*l;**

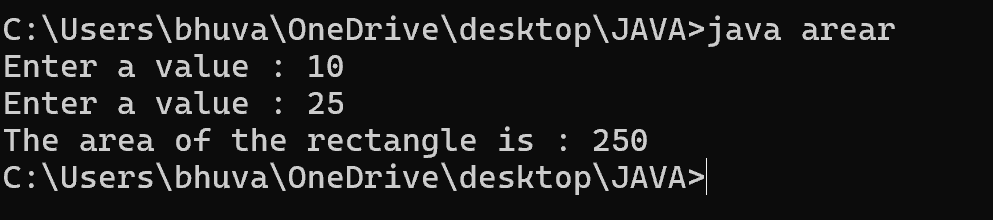
**System.out.print("The area of the rectangle is : "+ area);**

**input.close();**

**}**

**}**

**Output :**

****

**ERRORS :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **;** | **; is expected at end** |
| **2** | **area** | **Declaration of int type variable** |

1. **Write a java program to convert temperature from Celsius to Fahrenheit and vice versa.**

**Code : import java.util.Scanner;**

**class temp{**

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

**Scanner input =new Scanner(System.in);**

**System.out.print("enter the the temperature in degrees:");**

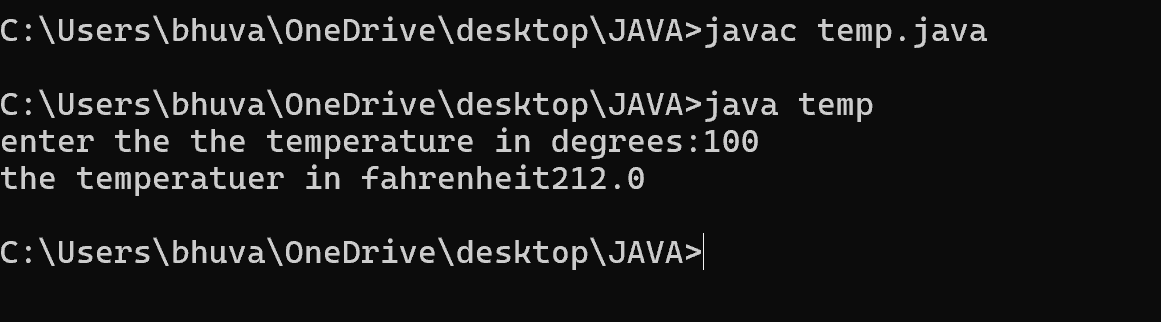
**double deg=input.nextDouble();**

**System.out.println("the temperatuer in fahrenheit"+((deg\*9/5)+32));**

**}**

**}**

**Output :**

****

**ERRORS :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **;** | **; is expected at end** |
| **2** | **Input.close();** | **The input is expected to be closed.** |

1. **Write a java program to calculate the simple interest.**

**Code : import java.util.Scanner;**

**public class si{**

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

**Scanner input = new Scanner(System.in);**

**System.out.print("Enter principal amount : ");**

**int p = input.nextInt();**

**System.out.print("Enter rate of interest : ");**

**int r = input.nextInt();**

**System.out.print("Enter the time period : ");**

**int t = input.nextInt();**

**int SI = p\*r\*t/100;**

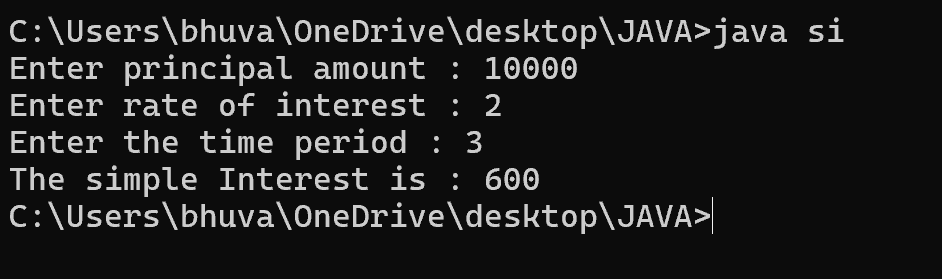
**System.out.print("The simple Interest is : " + SI);**

**input.close();**

**}**

**}**

**Output :**

****

**ERRORS :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **;** | **; is expected at end** |
| **2** | **Int t** | **Without declaring t the compiler cannot execute the program.** |

1. **Write a java program to find the largest of three numbers using ternary operation.**

**Code : import java.util.Scanner;**

**public class largest{**

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

**Scanner input = new Scanner(System.in);**

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

**int a = input.nextInt();**

**System.out.print("Enter number b : ");**

**int b = input.nextInt();**

**System.out.print("Enter number c : ");**

**int c = input.nextInt();**

**int largest = (a>=b) ? ((a>=c ) ? a : c) : ((b >=c) ? b : c);**

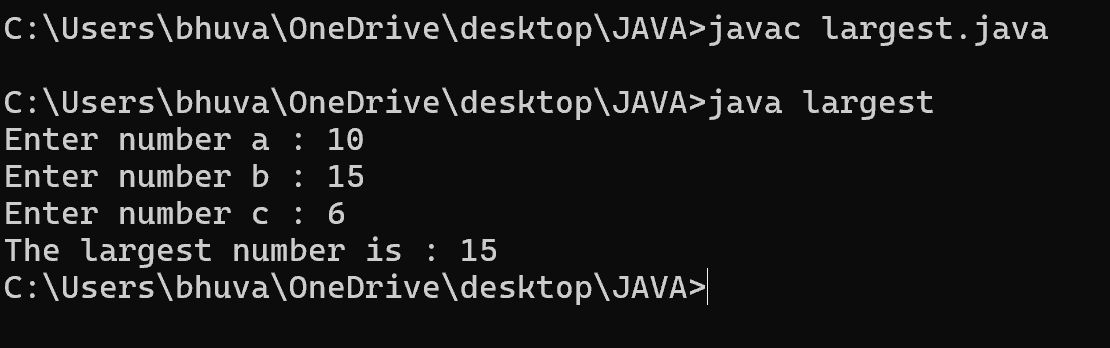
**System.out.print("The largest number is : " + largest);**

**input.close();**

**}**

**}**

**Output :**

****

**ERRORS :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **?** | **Checks the condition** |
| **2** | **:** | **Comparing between two variables** |

1. **Write a java program to find the factorial of a number**

**Code : import java.util.Scanner;**

**public class fac{**

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

**Scanner input = new Scanner(System.in);**

**System.out.print("Enter the number n : ");**

**int n = input.nextInt();**

**int fac = 1;**

**for(int i = 2; i<=n;i++){**

**fac \*= i;**

**}**

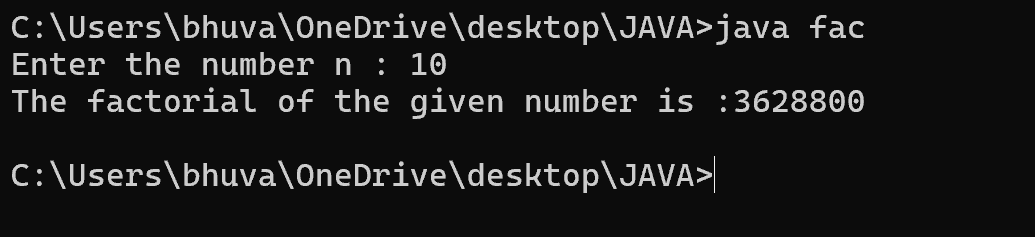
**System.out.println( "The factorial of the given number is :" + fac);**

**input.close();**

**}**

**}**

**Output :**

****

**ERRORS :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **}** | **To close for loop** |
| **2** | **System.out.print();** | **If we place the print statement inside the for loop it will print the each i value everytime but to print only the final value we must place it outside the for loop.** |

**WEEK – 3**

1. **Create the java program with the following instructions**
2. **Create a class with name Car**
3. **Create 4 attributes named Car\_Color , Car\_brand, fuel\_type, mileage**
4. **Create 3 method named Start( ) , Stop( ), Service( )**
5. **Create 3 objects Car1 , Car2 , Car3**
6. **Create a constructor which should print “Welcome to Car Garage”**

**Code: public class Car{**

**public String carColor;**

**private String carBrand;**

**private String fuelType;**

**public int mileage;**

**Car(String carColor , String carBrand , String fuelType , int mileage){**

**this.carColor = carColor;**

**this.carBrand = carBrand;**

**this.fuelType = fuelType;**

**this.mileage = mileage;**

**System.out.println(carColor + " " + carBrand + " " + fuelType + " " + mileage);**

**}**

**public void Start(){**

**System.out.println("The car has just started");**

**}**

**public void Stop(){**

**System.out.println("The car has just stopped");**

**}**

**public void Service(){**

**System.out.println("The car is in good condition");**

**}**

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

**Car Car1 = new Car("Black","Hyundai","Petrol",100);**

**Car Car2 = new Car("White","Suzuki","Diesel",150);**

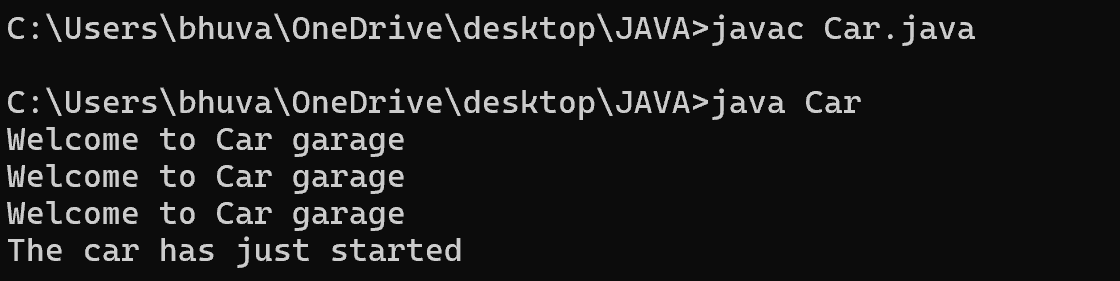
**Car Car3 = new Car("Red","Benz","Petrol",200);**

**Car1.Start();**

**}**

**}**

**Output :**

****

**Errors :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **}** | **} is expected at end of the class** |
| **2** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |

**Class Diagram :**

|  |
| --- |
| **Car** |
| **+ carColor : String**  **- carBrand : String**  **- fuelType : String**  **+ mileage : int** |
| **+ Car( ) : void**  **+ Start( ) : void**  **+ Stop( ) : void**  **+ Service( ) : void** |

1. **Write a java program to create a class BackAccount with two methods deposit( ) and withdraw( )**
2. **In deposit( ) whenever an amount is deposited it has to be updated with current amount**
3. **In withdraw( ) whenever an amount is withdrawn it has to be less than current amount else print “Insufficient funds”.**

**Code : public class BankAccount{**

**private String Name;**

**private int AccNo, CurrBal ;**

**BankAccount(String Name, int AccNo, int CurrBal){**

**this.Name = Name;**

**this.AccNo = AccNo;**

**this.CurrBal = CurrBal;**

**System.out.println("The customers are : " + this.Name + " ");**

**}**

**public int deposit(int dAmt){**

**CurrBal = CurrBal + dAmt ;**

**return CurrBal;**

**}**

**public void withdraw(int wAmount){**

**if(wAmount < CurrBal){**

**CurrBal = CurrBal - wAmount ;**

**System.out.println(CurrBal);**

**}**

**else{**

**System.out.println("Insufficient funds");**

**}**

**}**

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

**BankAccount Bhuvana = new BankAccount("Bhuvana",1500,10000);**

**Bhuvana.withdraw(13000);**

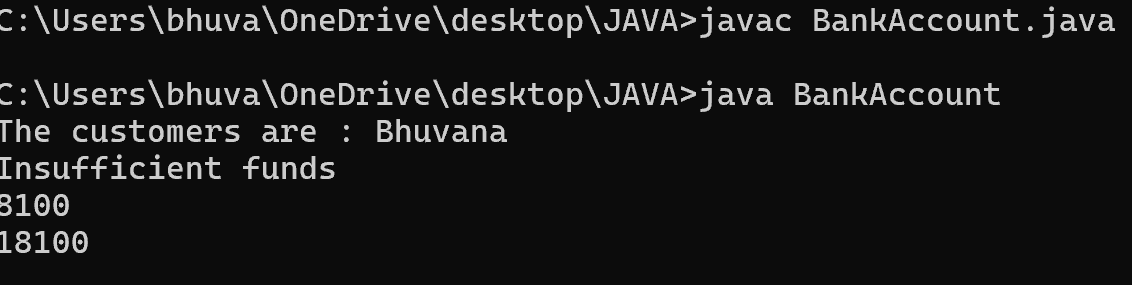
**Bhuvana.withdraw(1900);**

**int FinalAmount = Bhuvana.deposit(10000);**

**System.out.println(FinalAmount);**

**}**

**Output :**

****

**Errors :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **}** | **} is expected at end of the class** |
| **2** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |

**Class Diagram :**

|  |
| --- |
| **BankAccount** |
| **- Name : String**  **- AccNo : String**  **- CurrBal : String** |
| **+ BankAccount( ) : void**  **+ deposit( ) : int**  **+ withdraw( ) : void** |

**WEEK – 4**

1. **Write a java program with class named “Book”. The class should contain various attributes such as**

**“Title of the book , author , year of publication “. It should also contain a constructor with parameters**

**which initializes “ Title of the book, author, year of publication”. Create a method which displays the**

**details of the book. i.e. “ Title of the book, author and year of publication”. Display the details of two**

**books by creating two objects.**

**Code : class Book{**

**// beginning of the class book**

**public String Title;**

**private String author;**

**public int yearOfPublication;**

**// beginning of constructor**

**Book(String Title , String author , int yearOfPublication){**

**this.Title = Title;**

**this.author = author;**

**this.yearOfPublication = yearOfPublication;**

**}**

**//constructor ends here**

**// methos display starts here**

**public void display(){**

**System.out.println("Title of the book is : " + Title + "The name of the author is : " + author +**

**“The year of publication is : " + yearOfPublication );**

**}**

**// method display ends here**

**// creating objects**

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

**Book Book1 = new Book("Harry Potter" , "J.K.Rowling" ,1993);**

**Book Book2 = new Book("Someone Like You" , "Nikitha Singh" , 2010);**

**Book1.display();**

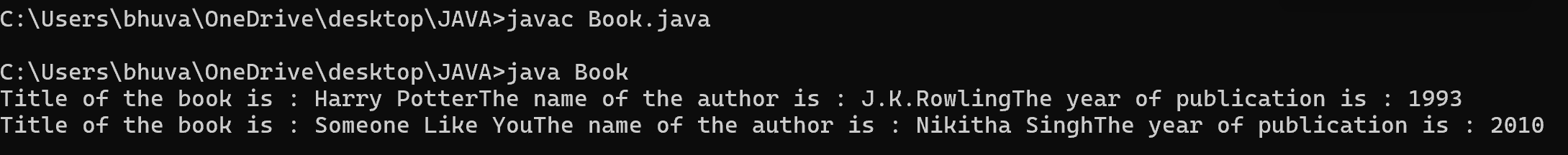
**Book2.display();**

**}**

**}**

**// class ends here**

**Output :**

****

**Errors :**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Expected Error** | **Reason** |
| **1** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |
| **2** | **}** | **Ending the class and main method is required** |

**Class Diagram :**

|  |
| --- |
| **Book** |
| **+ title : String**  **+ author : String**  **+ year : int** |
| **+ display() : void** |

1. **To create a java program with class named Myclass with a static variable “Count” of “int type”,**

**Initialized to 0 and a constant variable “pi” of type double , initialized to 3.1415 as attributes of that class**

**Now, define a constructor for “Myclass” that increments the “Count” variable each that an object of**

**Myclass is created. Finally , print the final values of “Count” and “pi” variables .**

**Code :**

**class Myclass{**

**// class starts here**

**static int Count = 0;**

**final double pi = 3.1415;**

**// the constructor starts here**

**Myclass(){**

**Count++;**

**}**

**// the constructor ends here**

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

**Myclass c1 = new Myclass();**

**Myclass c2 = new Myclass();**

**System.out.println("Count : " + c1.Count);**

**System.out.println("Pi : " + c1.pi);**

**}**

**}**

**// class ends here**

**Output :**

****

**Errors :**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Expected Error** | **Reason** |
| **1** | **.variable** | **We must mention variable name to call the variable** |
| **2** | **static** | **Static variables contain only one value** |

**Class Diagram :**

|  |
| --- |
| **myclass** |
| **+ static count : int=0**  **+ final pi : double=3.14** |
| **+ display() : void** |

**WEEK – 5**

1. **Create a calculator using the operations including addition, subtraction,multiplication,division using multilevel inheritance and display the desired output.**

**Code :**

**public class basiccalculator {**

**int a, b;**

**int sum, diff;**

**basiccalculator(int a, int b) {**

**this.a = a;**

**this.b = b;**

**}**

**public void add() {**

**sum = a + b;**

**System.out.println("Sum: " + sum);**

**}**

**public void diff(){**

**diff = a - b;**

**System.out.println("Difference: " + diff);**

**}**

**}**

**class advancecalc extends basiccalculator {**

**int mul;**

**advancecalc(int a, int b) {**

**super(a, b);**

**}**

**public void mult() {**

**mul = a \* b;**

**System.out.println("Multiplication: " + mul);**

**}**

**}**

**class aadvancecalc extends advancecalc {**

**float div;**

**aadvancecalc(int a, int b) {**

**super(a, b);**

**}**

**public void divi() {**

**if (b != 0) {**

**div = (float) a / b;**

**System.out.println("Division: " + div);**

**}**

**else {**

**System.out.println("Division by zero error!");**

**}**

**}**

**}**

**class ocalc {**

**public static void main(String[] args) { aadvancecalc c = new aadvancecalc(12, 6);**

**c.divi();**

**c.mult();**

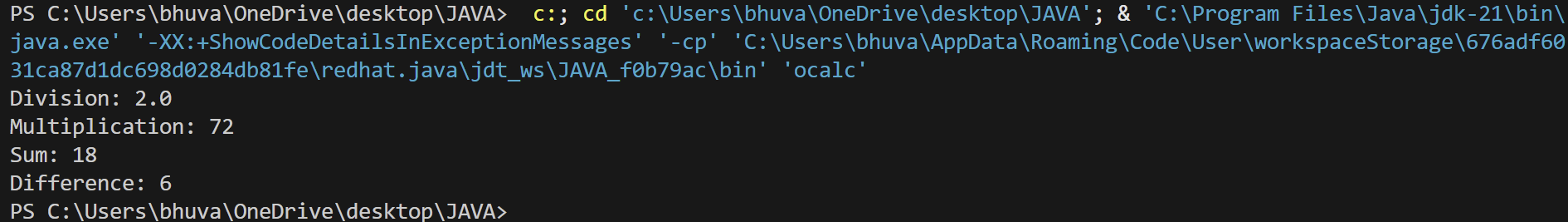
**c.add();**

**c.diff();**

**}**

**}**

**Output:**

****

|  |
| --- |
| **basiccalculator** |
| **+ a,b,sum,diff : int** |
| **+ add() : void**  **+bcalc()** |

**Class Diagram :**

****

|  |
| --- |
| **advancecalc** |
| **+ mul : int** |

|  |
| --- |
| **+ mult() : void**  **+acalc()** |

****

|  |
| --- |
| **aadvancecalc** |
| **+div: int** |
| **+ divi() : void**  **+aacalc()** |

1. **A vehicle rental company wants to develop a system that maintains information about different types**

**of vehicles available for rent. The company rents out cars and bikes and they need a program to store**

**details about each vehicle such as brand and speed**

**Cars should have an additional properties(attributes) - No. of doors , seating capacity**

**Bikes should have a property indicating whether they have gears or not**

**The system should also include a function to display details about each vehicle and indicate when a**

**vehicle is starting . Each class should have a constructor**

1. **Which oops concept is used in the above program ? Explain why is it useful in this scenario**
2. **If the company decides to add a new type of vehicle truck how would you modify the above**

**program**

1. **Truck should include an additional property called capacity(in tons)**
2. **Create a show truck details method to display the trucks capacity**
3. **Write a constructor for truck that initializes all the properties**
4. **Implement the truck class and update the main method to create the truck object and also create**

**an object for car and bike subclass. Finally display its details**

**Code :**

**class Vehicle {**

**String brand; int speed;**

**Vehicle(String brand, int speed) {**

**this.brand = brand; this.speed = speed;**

**}**

**void displayDetails() {**

**System.out.println("Brand: " + brand); System.out.println("Speed: " + speed + " km/h");**

**}**

**void startVehicle() {**

**System.out.println(brand + " is starting...");**

**}**

**}**

**// Car subclass**

**class Car1 extends Vehicle { int noOfDoors;**

**int seatingCapacity;**

**Car1(String brand, int speed, int noOfDoors, int seatingCapacity) { super(brand, speed);**

**this.noOfDoors = noOfDoors;**

**this.seatingCapacity = seatingCapacity;**

**}**

**@Override**

**void displayDetails() { super.displayDetails();**

**System.out.println("Number of Doors: " + noOfDoors); System.out.println("Seating Capacity: " + seatingCapacity);**

**}**

**}**

**// Bike subclass**

**class Bike extends Vehicle { boolean hasGears;**

**Bike(String brand, int speed, boolean hasGears) { super(brand, speed);**

**this.hasGears = hasGears;**

**}**

**@Override**

**void displayDetails() { super.displayDetails();**

**System.out.println("Has Gears: " + (hasGears ? "Yes" : "No"));**

**}**

**}**

**// Truck subclass**

**class Truck extends Vehicle { double capacity;**

**Truck(String brand, int speed, double capacity) { super(brand, speed);**

**this.capacity = capacity;**

**}**

**void showTruckDetails() {**

**System.out.println("Truck Capacity: " + capacity + " tons");**

**}**

**@Override**

**void displayDetails() { super.displayDetails(); showTruckDetails();**

**}**

**}**

**// Main class**

**public class VehicleRentalSystem {**

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

**Car1 car = new Car1("Mercedes Benz", 150, 4, 5);**

**Bike bike = new Bike("Yamaha", 120, true); Truck truck = new Truck("Volvo", 100, 15.5);**

**System.out.println("Car Details:"); car.displayDetails();**

**car.startVehicle(); System.out.println();**

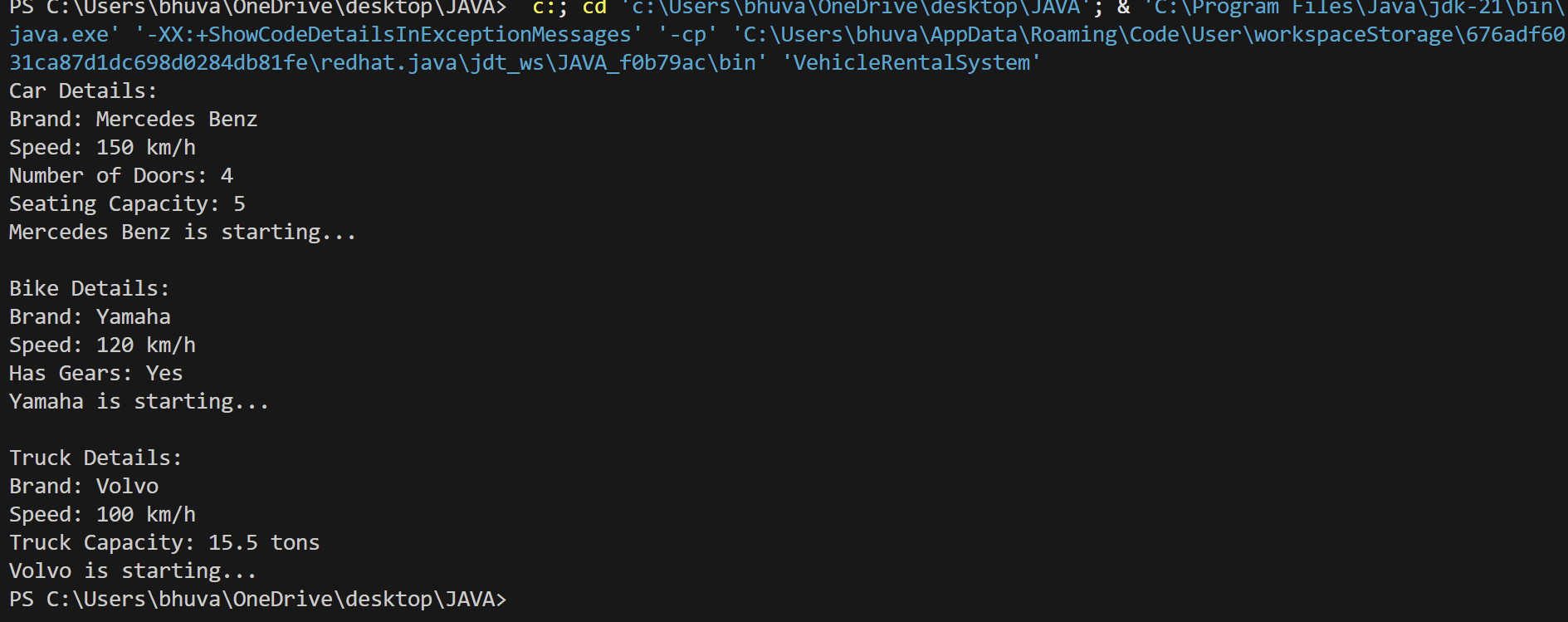
**System.out.println("Bike Details:"); bike.displayDetails();**

**bike.startVehicle(); System.out.println();**

**System.out.println("Truck Details:"); truck.displayDetails();**

**truck.startVehicle();**

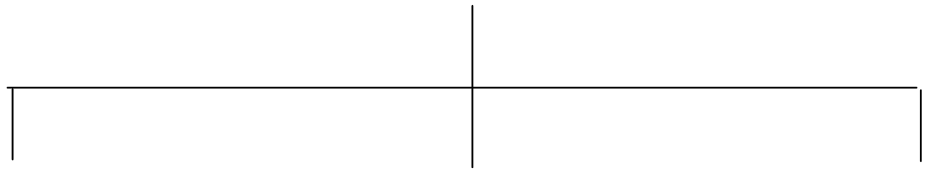
**Output :**

****

|  |
| --- |
| **Vehicle** |
| **+brand : String**  **+speed : int** |
| **+Vehicle()**  **+displayDetails():void**  **+startVehicle():void** |

**Class Diagram :**

|  |
| --- |
| **Car** |
| **+noOfDoors : int**  **+seatingCapacity : int** |
| **+Car()**  **+displayDetails():void** |

****

|  |
| --- |
| **Bike** |
| **+ hasGears : boolean** |
|  |
| **+Bike()**  **+displayDetails():void** |

|  |
| --- |
| **Truck** |
| **+capacity : double** |
| **+Truck()**  **+showTruckDetails():void**  **+displayDetails():void** |

**WEEK – 6**

1. **Write a java program to create a Vehicle class with displayInfo() method , overridden in Car subclass to provide info about carcompany , model , price, seating and petrol.**

**Code :**

**class Vehicle2{**

**public void displayInfo(String comp,String model,int price,int**

**seating,boolean petrol){**

**System.out.println("Details");**

**}**

**}**

**class car extends Vehicle2{**

**public void displayInfo(String comp,String model,int price,int**

**seating,boolean petrol){**

**System.out.println("Car Details");**

**System.out.println("Car company:"+comp);**

**System.out.println("Car model:"+model);**

**System.out.println("Car seating:"+seating);**

**System.out.println("Car price:"+price);**

**System.out.println("Petrol:"+petrol);**

**}**

**}**

**class Main{**

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

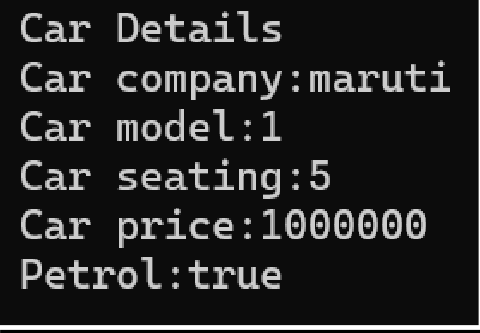
**car c=new car();**

**c.displayInfo("maruti","1",1000000,5,true);**

**}**

**}**

**Output :**

****

**Class Diagram :**



|  |
| --- |
| Vehicle |
| +displayInfo(): void |

|  |
| --- |
| car |
| +displayInfo(): void |

**Errors :**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Expected Error** | **Reason** |
| **1** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |
| **2** | **}** | **Ending the class and main method is required** |

1. **An automated admission system that verifies student eligibility for UG and PG with different criteria.**

**.UG requires minimum of 60%**

**.PG requires minimum of 70**%

**Code :**

**class adm{**

**public void elg(float score){ System.out.println("Eligibility");**

**}**

**}**

**class ug extends adm{**

**public void elg(float score){**

**if(score>=60){**

**System.out.println("Eligible");**

**}**

**else{**

**System.out.println("Not Eligible");**

**}**

**}**

**}**

**class pg extends Admission{**

**public void elg(float score){**

**if(score>=70){**

**System.out.println("Eligible");**

**}**

**else{**

**System.out.println("Not Eligible");**

**}**

**}**

**}**

**class eligibility{**

**public static void main(String[] args){ ug stu1=new ug();**

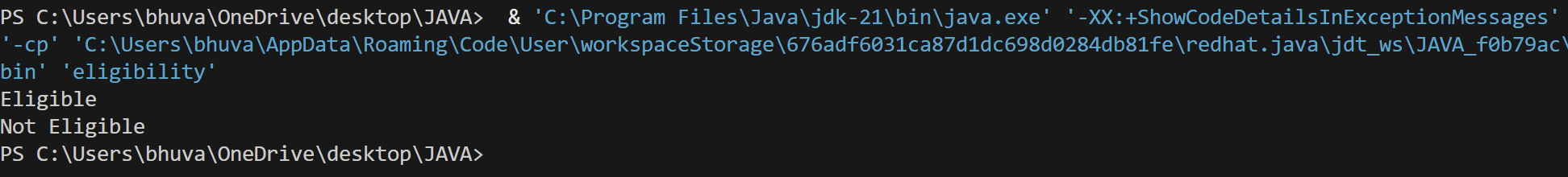
**pg stu2=new pg(); stu1.elg(94);**

**stu2.elg(69);**

**}**

**}**

**Output :**

****

**Class Diagram :**

|  |
| --- |
| **adm** |
| **+elg():void** |

|  |
| --- |
| **ug** |
| **+elg():void** |

|  |
| --- |
| **pg** |
| **+elg():void** |

**Errors :**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Expected Error** | **Reason** |
| **1** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |
| **2** | **}** | **Ending the class and main method is required** |

1. **Create a calculator class with overloaded methods to perform additions**

**.add two integers**

**.add two double values**

**.add three integers**

**Code :**

**public class cal {**

**public int add(int a,int b){**

**return a+b;**

**}**

**public double add(double a, double b){**

**return a+b;**

**}**

**public int add(int a,int b,int c){ return a+b+c;**

**}**

**}**

**class result{**

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

**cal c=new cal();**

**System.out.println(c.add(2,3));**

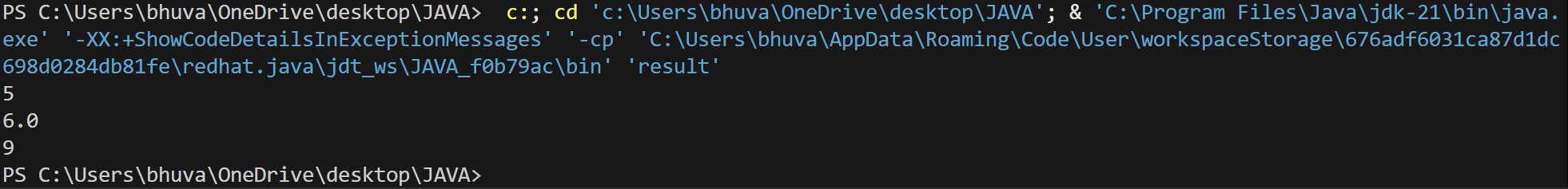
**System.out.println(c.add(2.5,3.5));**

**System.out.println(c.add(2,3,4));**

**}**

**}**

**Output :**

****

**Class Diagram :**

|  |
| --- |
| **cal** |
| **+add(int a,int b):int**  **+add(double a,double b):double**  **+add(int a,int b,int c):int** |

**Errors :**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Expected Error** | **Reason** |
| **1** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |
| **2** | **}** | **Ending the class and main method is required** |

1. **Create a shape class with method calculateArea() that is overloaded for different shapes (eg: square, rectangle).Then create a subclass Circle that overrides calculateArea() method for Circle.**

**Code :**

**public class sha {**

**public float calarea(float side){**

**return side\*side;**

**}**

**public float calarea(float l,float b){**

**return l\*b;**

**}**

**}**

**class circles extends sha{**

**public double calarea(double r){**

**return 3.14\*r\*r;**

**}**

**}**

**class s{**

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

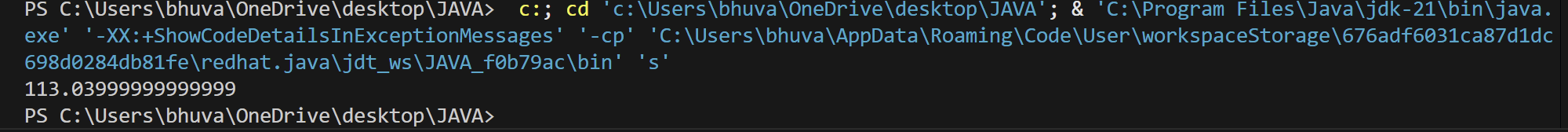
**circles c=new circles();**

**System.out.println(c.calarea(6.0));**

**}**

**}**

**Output :**

****

**Class Diagram :**

|  |
| --- |
| **shape** |
| **+calarea(float side):float**  **+calarea(float l,float b):float** |

****

|  |
| --- |
| **circle** |
| **+calarea(double r):double** |

**Errors :**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Expected Error** | **Reason** |
| **1** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |
| **2** | **}** | **Ending the class and main method is required** |

**WEEK – 7**

**1) Write a java program to create an abstract class Animal with an abstract method called sound( ).**

**Create subclasses Lion and Tiger that extend the Animal cass and implement the sound( ) method to**

**make a specific sound for each animal.**

**Code :**

**abstract class Animal {**

**abstract void sound();**

**}**

**class Lion extends Animal{**

**public void sound(){**

**System.out.println("The lion is roaring..");**

**}**

**}**

**class Tiger extends Animal {**

**public void sound(){**

**System.out.println("The tiger is roaring");**

**}**

**}**

**class Main1{**

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

**Lion a = new Lion();**

**a.sound();**

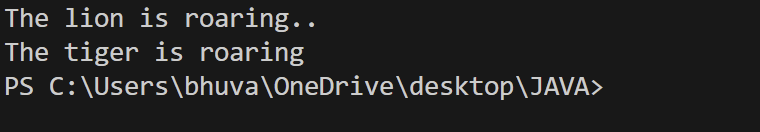
**Tiger b = new Tiger();**

**b.sound();**

**}**

**}**

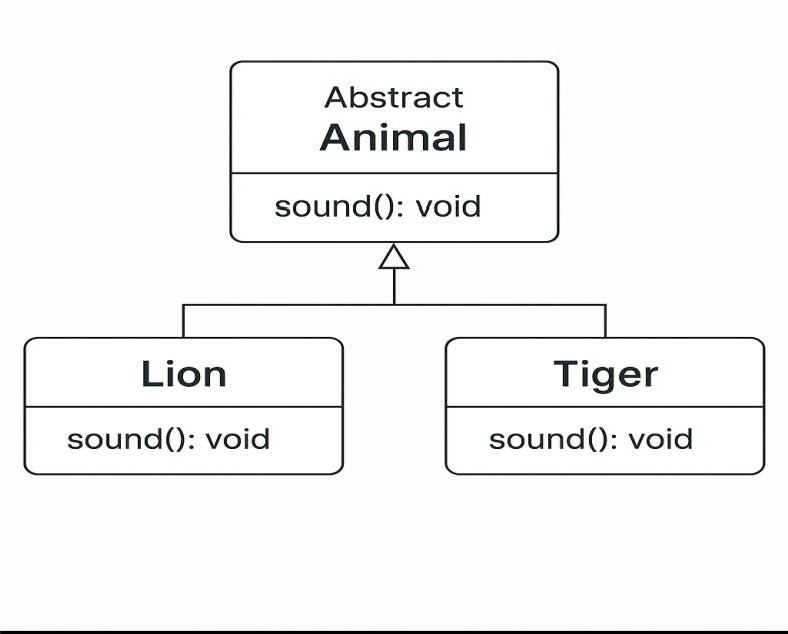
**Output :**

****

**Errors :**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Expected Error** | **Reason** |
| **1** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |
| **2** | **}** | **Ending the class and main method is required** |

**Class Diagram :**

****

**2) Write a java program to create an abstract class shape3D with abstract methods to calculate volume**

**and** **surfacearea and create subclasses for sphere and cube that implements these methods.**

**Code :**

**abstract class Shape3D{**

**double radius;**

**double side;**

**abstract void calculateVolume(double radius);**

**abstract void calculateSurfaceArea(double radius);**

**}**

**class Sphere extends Shape3D{**

**final double pi = 3.14;**

**double volume;**

**double surfaceArea;**

**public void calculateVolume(double radius){**

**this.radius = radius;**

**volume =  4.0/3.0\*pi\*radius\*radius\*radius;**

**System.out.println("The volume of the sphere is : " + volume);**

**}**

**public void calculateSurfaceArea(double radius){**

**this.radius = radius;**

**surfaceArea = (4\*pi\*radius\*radius);**

**System.out.println("The surface area of the sphere is : " + surfaceArea);**

**}**

**}**

**class Cube extends Shape3D{**

**double volume;**

**double surfaceArea;**

**public void calculateVolume(double radius){**

**this.side =radius;**

**volume =side\*side\*side;**

**System.out.println("The volume of the cube is : " + volume);**

**}**

**public void calculateSurfaceArea(double radius){**

**this.side = radius;**

**surfaceArea = 6\*side\*side;**

**System.out.println("The surface area of the cube is :"+ surfaceArea);**

**}**

**}**

**class Main3{**

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

**Sphere s = new Sphere();**

**s.calculateVolume(6.0);**

**s.calculateSurfaceArea(5.5);**

**Cube c = new Cube();**

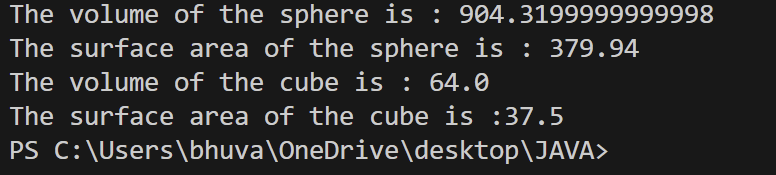
**c.calculateVolume(4.0);**

**c.calculateSurfaceArea(2.5);**

**}**

**}**

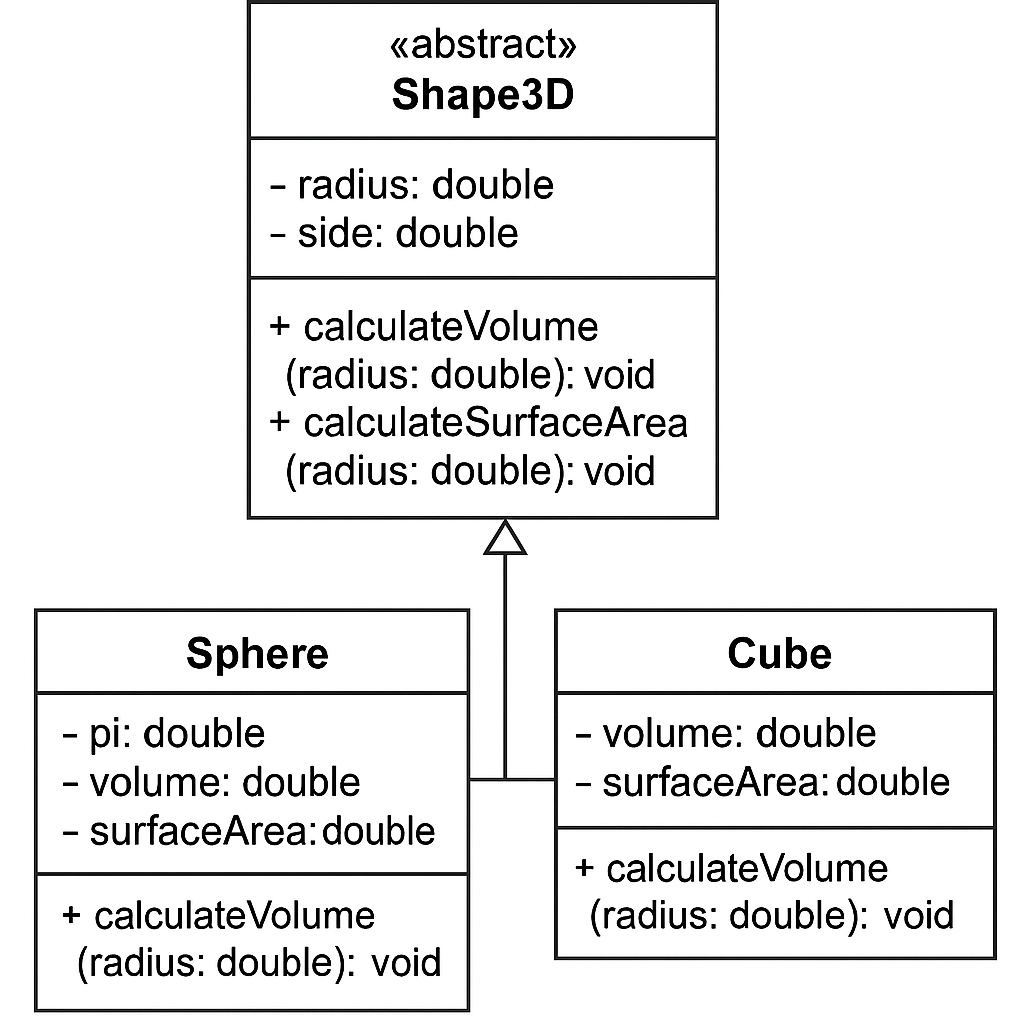
**Output :**

****

**Error :**

|  |  |  |
| --- | --- | --- |
| S.No. | Expected Error | Reason |
| 1 | Setting the parameters inside the constructor | We cannot pass the values inside constructor without setting them first |
| 2 | } | Ending the class and main method is required |

**Class Diagram :**

****

**3) Create an abstract class PatternPrint with an abstract method printing to print the pattern and a concrete**

**method to to display the pattern . Implement the patterns**

1. **Star Pattern - prints a right angled triangle of stars**
2. **Number Pattern – prints a right angled triangle of increasing numbers.**

**Code :**

**import java.util.Scanner;**

**abstract class Printpattern {**

**public abstract void printPattern(int n);**

**}**

**class RightTrianglePattern extends Printpattern {**

**@Override**

**public void printPattern(int n) {**

**System.out.println("Right Triangle Pattern:");**

**for (int i = 1; i <= n; i++) {**

**for (int j = 1; j <= i; j++) {**

**System.out.print("\* ");**

**}**

**System.out.println();**

**}**

**}**

**}**

**class NumberPattern extends Printpattern {**

**@Override**

**public void printPattern(int n) {**

**System.out.println("number pattern:");**

**for (int i =1; i <= n; i++) {**

**for (int j = 1; j <= i; j++) {**

**System.out.print( j);**

**}**

**System.out.println();**

**}**

**}**

**}**

**class pattern  {**

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

**Scanner input= new Scanner(System.in);**

**System.out.println("enter the n value to select number of rows");**

**int n=input.nextInt();**

**Printpattern rightTriangle = new RightTrianglePattern();**

**Printpattern numberpattern = new NumberPattern();**

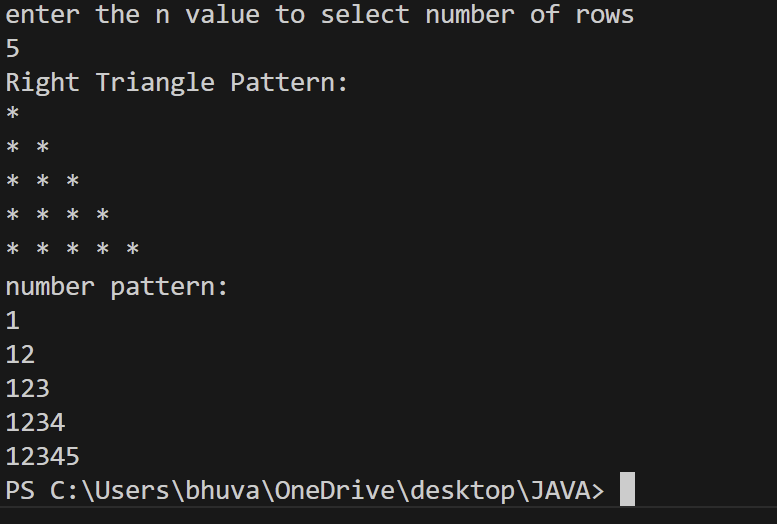
**rightTriangle.printPattern(n);**

**numberpattern.printPattern(n);**

**}**

**}**

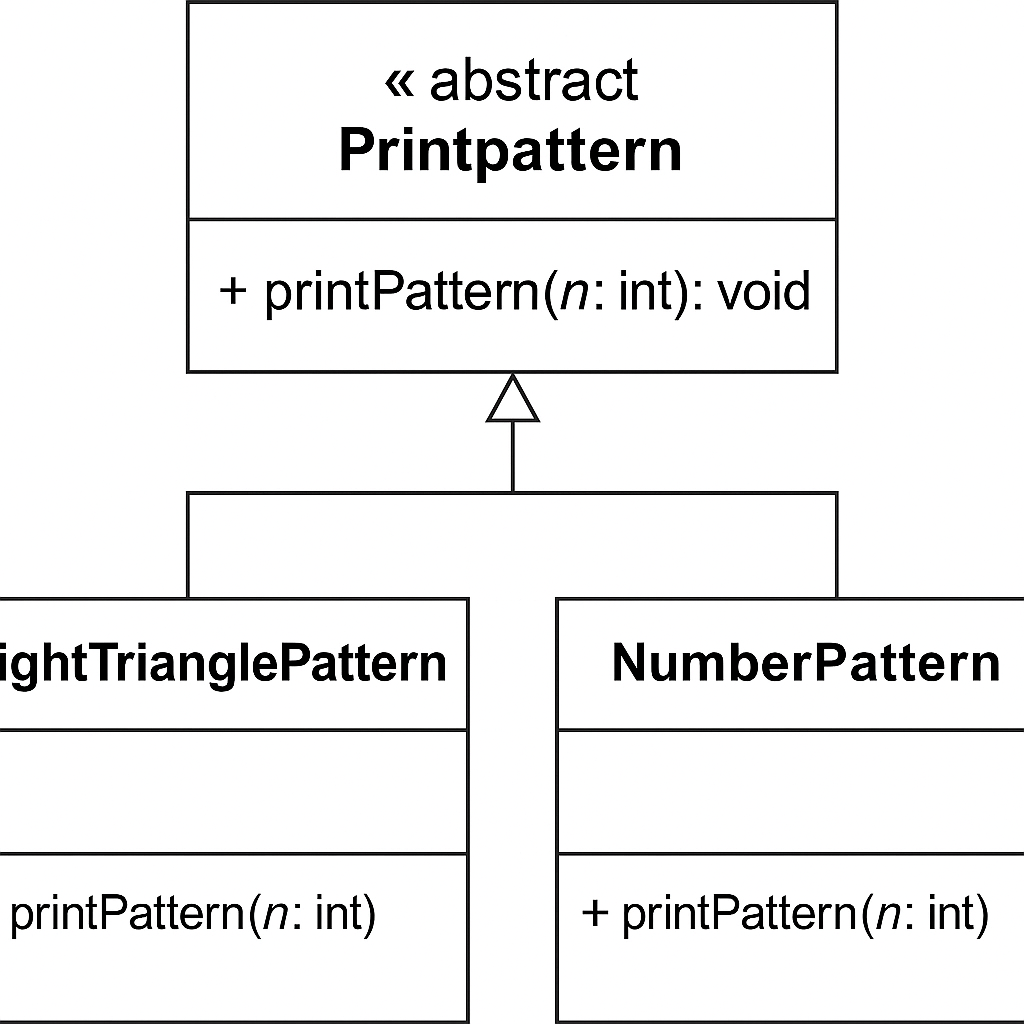
**Output :**

****

**Error :**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Expected Error** | **Reason** |
| **1** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |
| **2** | **}** | **Ending the class and main method is required** |

**Class Diagram :**

****