**23CSE111**

**Object Oriented Programming**



**Department of Computer Science Engineering**

**Amrita School of Engineering**

**Amrita Vishwa Vidyapeetham, Amaravati Campus**

**Name: Bhargavi.CH**

**Verified By : Roll No: AV.SC.CSE24101**

**JAVA:**

Java is a high level, class based, object oriented programming language that is widely used across various operating systems.

**Installation of JAVA in Computer:**

**Aim:**

Solving computational programming using JAVA.

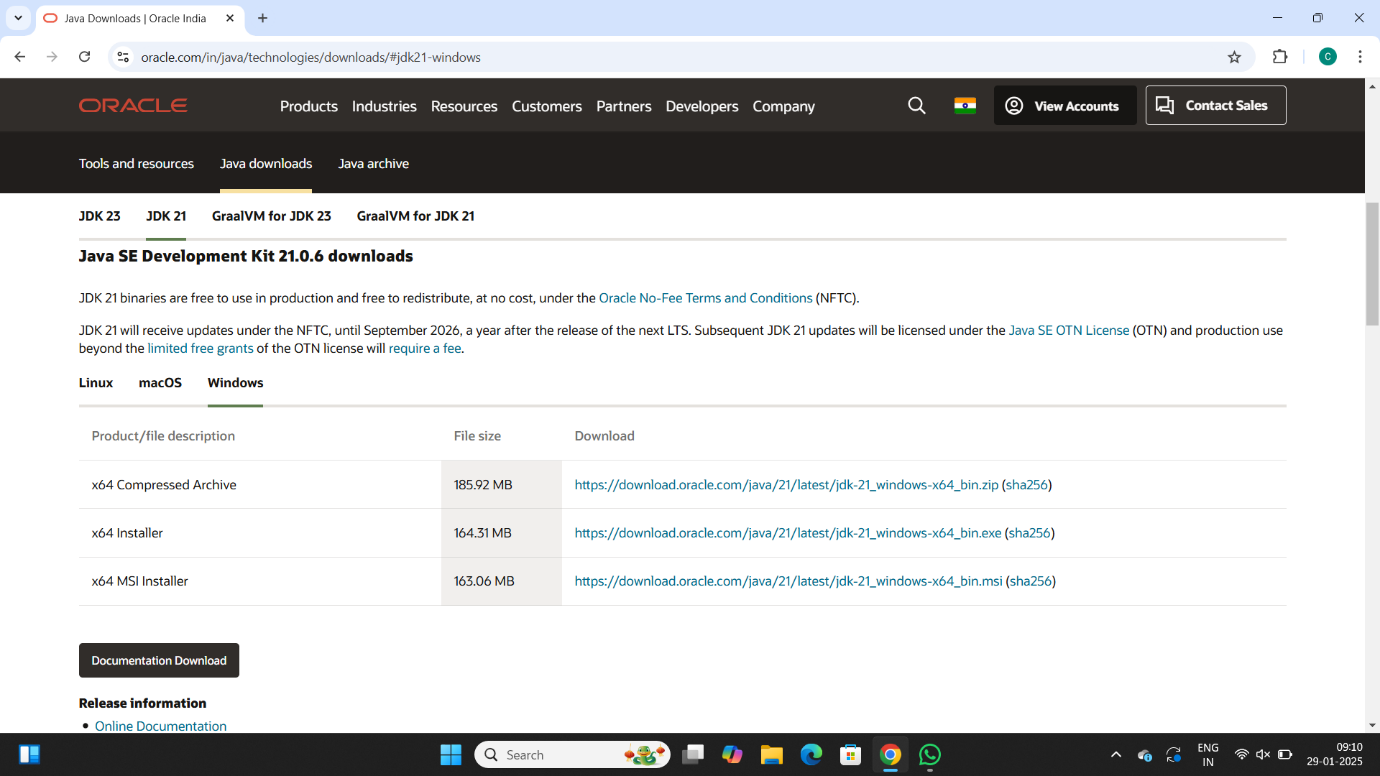
**Procedure:**

Follow the steps below to install Java on Windows:

1. Download JDK(Java Development Kit)
2. Run the Installer
3. Configure Environment Variables
4. Verify Installation

1.Download JDK:

Go to the official oracle website to download the JDK. Choose x64 MSI Installer on the windows tab and click on download link.



2. Run the Installer:

Now, go to your downloads folder and run the installer you just downloaded.

3. Configure Environment Variables:

After installation, you will need to tell your system where to find Java. This is done by setting environment variables.

Locate JDK Path:

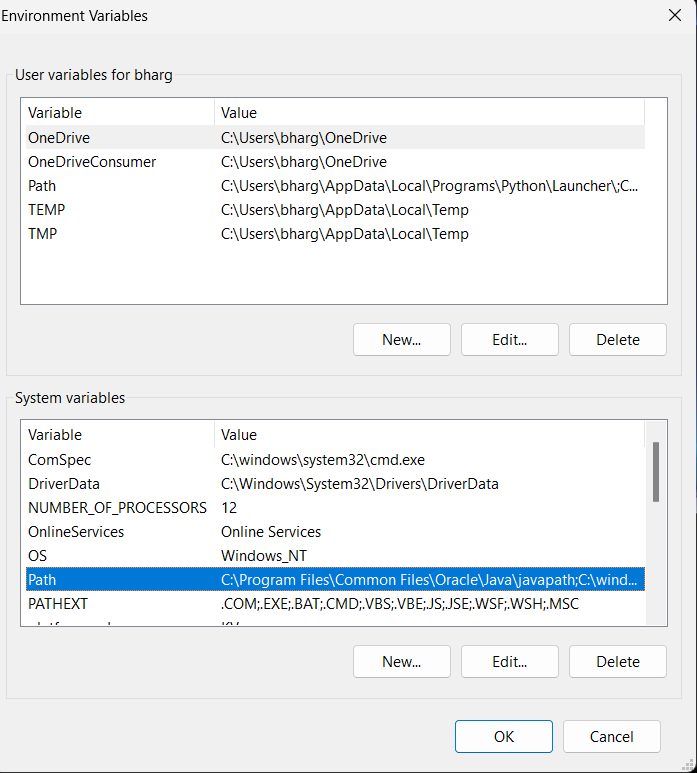
Navigate through your file explorer to reach the JDK installation directory. Normally, it is located at -

C:\Program Files\Java\jdk-22\bin

Copy this path.

Access Environment Variables:

Search environment variable on the terminal. In system properties, click on environment variables. You will be prompted to the screen below.



Update the Path Variable:

Find the Path variable in the System variables section and click on edit. Then, click New and paste your JDK bin path.

Finally, click Ok to close each window.

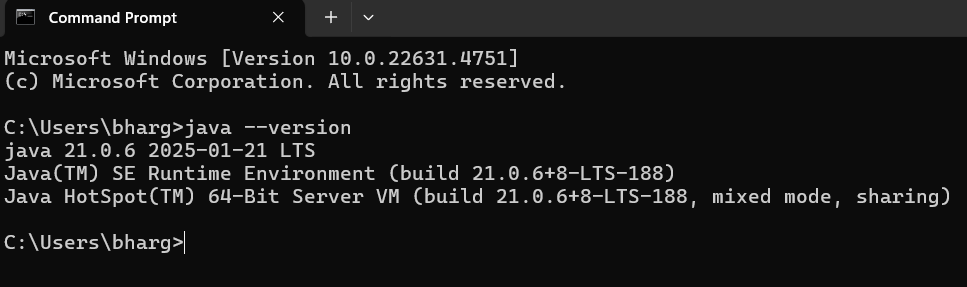
Set JAVA\_HOME Variable:

Back in the environment variables window, under the system variables section, click New to create a new variable.

Now, name the variable  and set its value to the path of the JDK folder directory. Close all the dialogues with the Ok button.

1. Verify Installation:

After the installation, you can verify whether Java is installed by using the following command in the command prompt.



If Java is installed successfully, it will print the version information; otherwise, it will produce an error message indicating that the command is not recognized.

**JAVA Program:**

**INPUT:**

Class program{

Public static void main(string[] args){

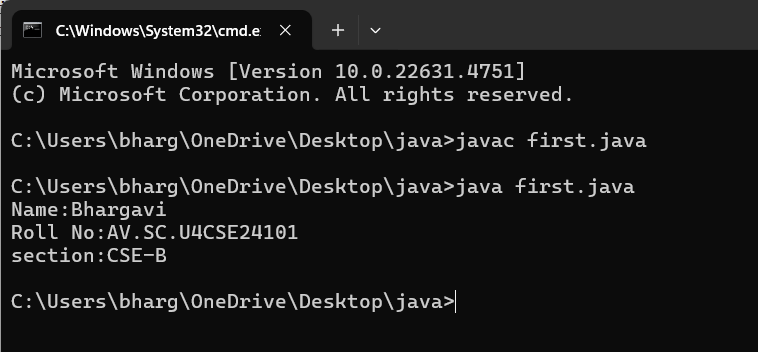
System.out.println(“Name:Bhargavi.ch”);

System.out.println(“Section:CSE-B”);

System.out.println(“Roll No:AV.SC.U4CSE24101”);

}

}



**WEEK 02**

**PROGRAM-1:**

**AIM:** Write a Java program to calculate area of rectangle.

**CODE:**

import java.util.Scanner;

class Area {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter length: ");

float l = input.nextFloat();

System.out.print("Enter width: ");

float b = input.nextFloat();

input.close();

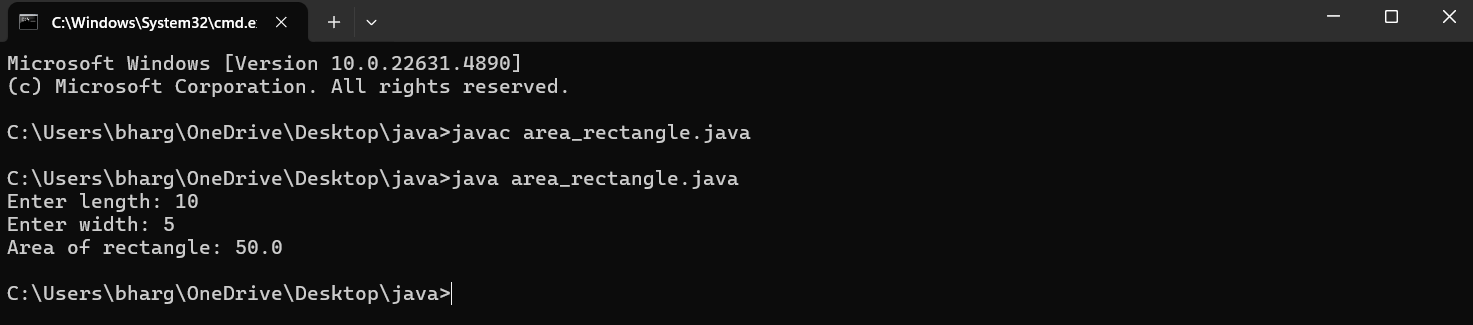
float area = l \* b;

System.out.println("Area of rectangle: " + area);

}

}

**OUTPUT:**



**PROGRAM-2:**

**AIM:** Write a Java program to convert temperature from Fahrenheit to Celsius and vice versa.

**CODE:**

import java.util.Scanner;

class Temperature {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter Temperature in Fahrenheit: ");

float F = input.nextFloat();

input.close();

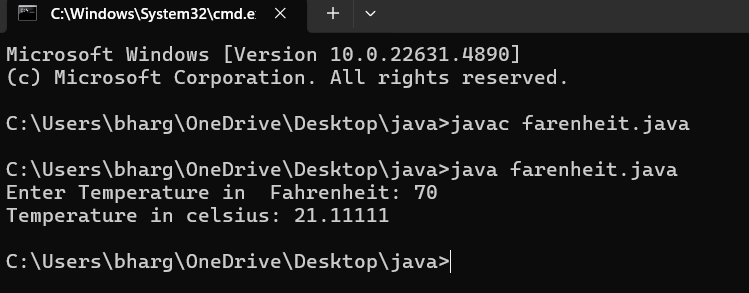
float C = (F - 32)\*5/9;

System.out.println("Temperature in celsius: " + C);

}

}

**OUTPUT:**



**CODE:**

import java.util.Scanner;

class temperature2 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter Temperature in Celsius: ");

float C = input.nextFloat();

input.close();

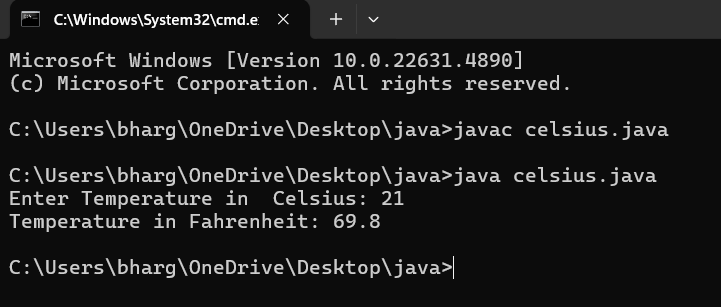
float F = (C \* 9/5) + 32;

System.out.println("Temperature in Fahrenheit: " + F);

}

}

**OUTPUT:**

****

**PROGRAM-3:**

**AIM:** Write a Java program to calculate simple intrest.

**CODE:**

import java.util.Scanner;

class simple\_interest {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter principle: ");

float P = input.nextFloat();

System.out.print("Enter time: ");

float T = input.nextFloat();

System.out.print("Enter rate: ");

float R = input.nextFloat();

input.close();

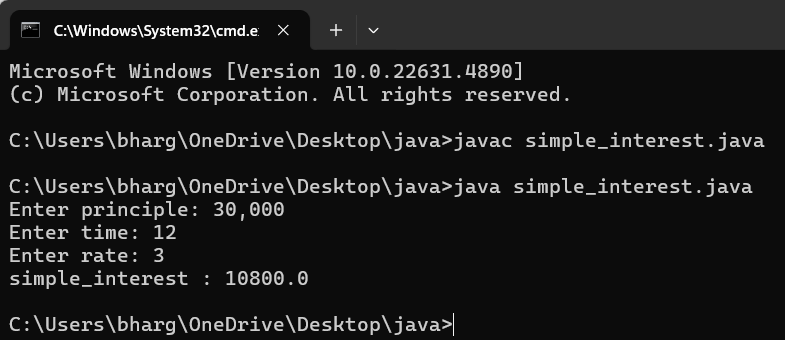
float SI = (P\*T\*R)/100;

System.out.println("simple\_interest : " + SI );

}

}

**OUTPUT:**

****

**PROGRAM-4:**

**AIM:** Write a Java program to calculate largest of 3 numbers using ternary operators.

**CODE:**

import java.util.Scanner;

class largest\_num {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter number1: ");

float n1 = input.nextFloat();

System.out.print("Enter number2: ");

float n2 = input.nextFloat();

System.out.print("Enter number3: ");

float n3 = input.nextFloat();

input.close();

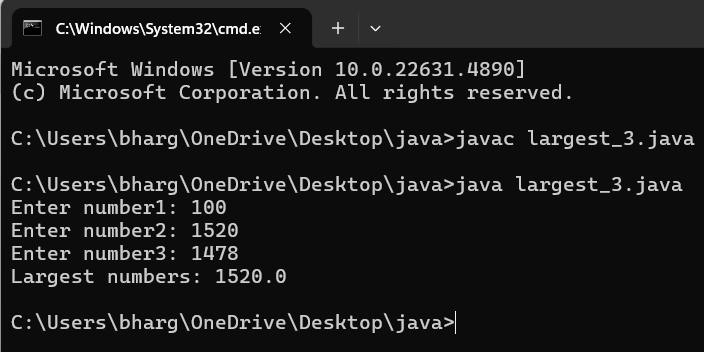
float largest = (n1 >= n2) ? ((n1 >= n3) ? n1 : n3) : ((n2 >= n3) ? n2 : n3);

System.out.println("Largest numbers: " +largest);

}

}

**OUTPUT:**

****

**PROGRAM-5:**

**AIM:** Write a Java program to calculate factorial of a number.

**CODE:**

import java.util.Scanner;

class factorial {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int n = input.nextInt();

input.close();

long factorial = 1;

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

factorial \*= i;

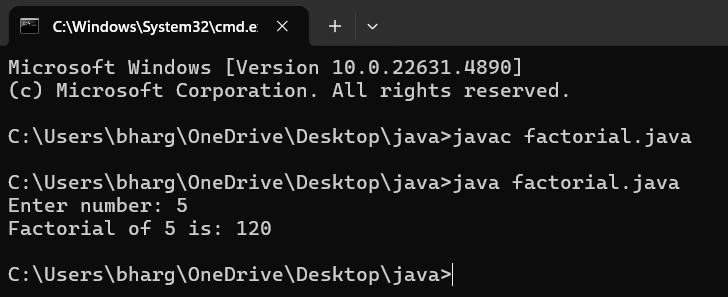
}

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

}

}

**OUTPUT:**



**WEEK – 3:**

**PROGRAM- 1:**

**AIM:** To create a class BankAccount with methods deposit() and withdraw() . create two subclasses savingsaccount and checkingaccount override the withdraw () method in each subclass to impose different withdrawal limits and fees.

**CODE:**

public class Bankaccount {

public String accountHolder;

public double balance;

public int accountNumber;

public Bankaccount(String accountHolder, int accountNumber, double balance) {

this.accountHolder = accountHolder;

this.accountNumber = accountNumber;

this.balance = balance;

}

public void withdrawal(double amount) {

if (amount <= balance) {

balance -= amount;

System.out.println("Current balance: " + balance);

} else {

System.out.println("Invalid withdrawal amount");

}

}

public void deposit(double amount) {

balance += amount;

System.out.println("Current balance: " + balance);

}

public static void main(String[] args) {

Bankaccount ba = new Bankaccount("Ch.Bhargavi", 24151, 1000);

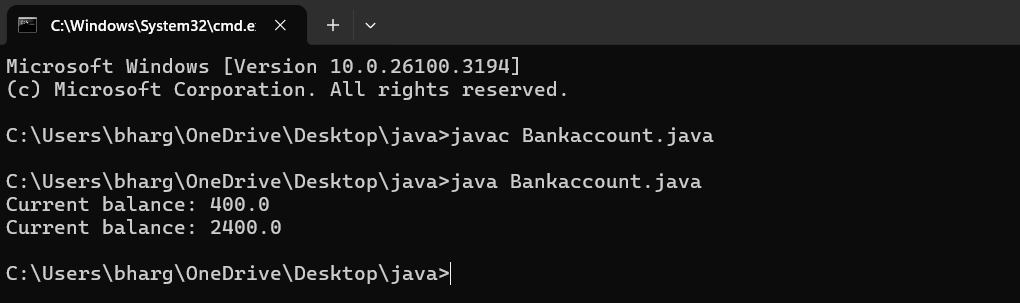
ba.withdrawal(600);

ba.deposit(2000);

}

}

Output:



ERROR TABLE:

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Not putting the semi-colon; after calling the function. 2. After Withdrawal,deposit not giving the parenthesis ( ). | 1. Put the semi-colon after the writing the code. 2. After every method, put the parenthesis ( ). |

IMPORTANT POINTS:

1. The condition inside the if statement must be correct.
2. It explains that if the withdrawal money is less than the money in the bank account, then we can withdraw the amount.

**Class diagram:**

|  |
| --- |
| **BankAccount**  **----------------------------------------------------------**  **-balance: double**  **----------------------------------------------------------**  **+BankAccount(intialBalance: double)**  **+deposit(amount: double):void**  **+withdraw(amount: double):void** |

**PROGRAM- 2:**

**AIM:** To create java program with following instructions :

1. Create a class with name Car

2. Create four attributes named car\_color, car\_brand, fuel\_type, mileage

3. Create these methods named start(),stop(),service()

4. Create the objects named car, car1,car2.

**CODE:**

public class car {

private String car\_color;

private String car\_brand;

private String fuel\_type;

private String mileage;

public void start() {

System.out.println("car is started");

}

public void stop() {

System.out.println("car is stopped");

}

public void service() {

System.out.println("car is for service");

}

public static void main(String args[]) {

car car = new car();

car.car\_color = "blue";

car.car\_brand = "wolkswagen";

car.fuel\_type = "petrol";

car.mileage = "40";

car.start();

System.out.println("car\_color: " + car.car\_color + " car\_brand: " + car.car\_brand + " fuel\_type: " + car.fuel\_type + " mileage: " + car.mileage);

car car\_one = new car();

car\_one.car\_color = "grey";

car\_one.car\_brand = "maruti suzuki";

car\_one.fuel\_type = "petrol";

car\_one.mileage = "75";

car\_one.stop();

System.out.println("car\_color: " + car\_one.car\_color + " car\_brand: " + car\_one.car\_brand + " fuel\_type: " + car\_one.fuel\_type + " mileage: " + car\_one.mileage);

car car\_two = new car();

car\_two.car\_color = "white";

car\_two.car\_brand = "Mercedes benz";

car\_two.fuel\_type = "diesel";

car\_two.mileage = "45";

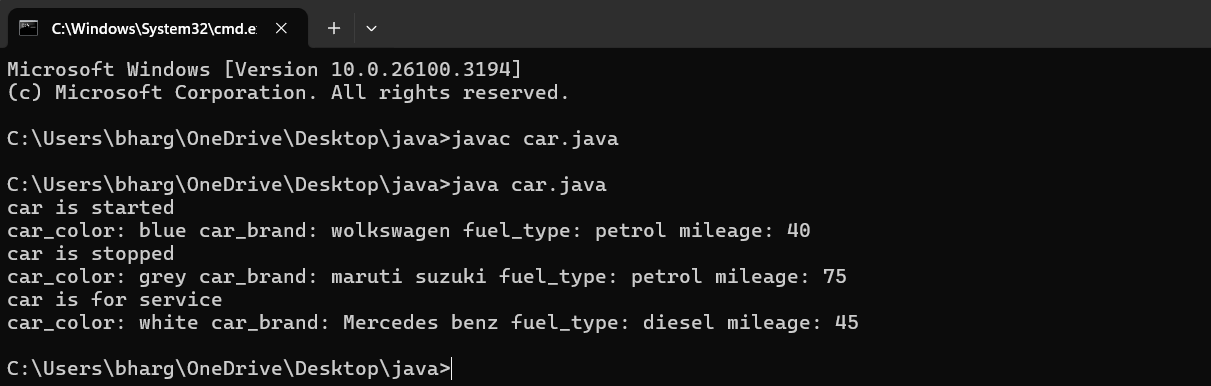
car\_two.service();

System.out.println("car\_color: " + car\_two.car\_color + " car\_brand: " + car\_two.car\_brand + " fuel\_type: " + car\_two.fuel\_type + " mileage: " + car\_two.mileage);

}

}

Output:



ERROR TABLE:

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Not putting the semi-colon; after calling the function. 2. After Start, Stop, Service not giving the parenthesis ( ). | 1. Put the semi-colon after the writing the code. 2. After every method, put the parenthesis ( ). |

IMPORTANT POINTS:

1. Before calling the function we should write the method properly.
2. Here, the “public void start( )” indicates that we are writing a method to call the function.
3. When we call a certain method, the process inside it will be printed as an output of the code.
4. Here the details inside the function are called objects, we can give any objects

**Class diagram:**

|  |
| --- |
| **car**  **-----------------------**  **-car\_color:string**  **-car\_brand:string**  **-fuel\_type:string**  **-milage:double**  **----------------------**  **+start():void**  **+stop():void**  **+service():void** |

**WEEK -4**

**PROGRAM – 1:**

**AIM**: Write a java program with class named “book”, the class should contain various attributes such as title, author, year of publication it should also contain a constructor with parameters which initializes, title, author, and year of publication.

Create a method which displays the details of the book and display the details of two books.

**CODE:**

public class Book{

public String Title;

public String Author;

public String Year\_of\_publication;

public Book(String Title, String Author, String Year\_of\_publication){

this.Title=Title;

this.Author=Author;

this.Year\_of\_publication=Year\_of\_publication;

}

public void Bookdetails()

{

System.out.println("Title:"+Title);

System.out.println("Author:"+Author);

System.out.println("Year\_of\_publication:"+Year\_of\_publication);

}

public static void main (String[] args){

Book book1=new Book("Mahabaratam","Vyasa","3rd century");

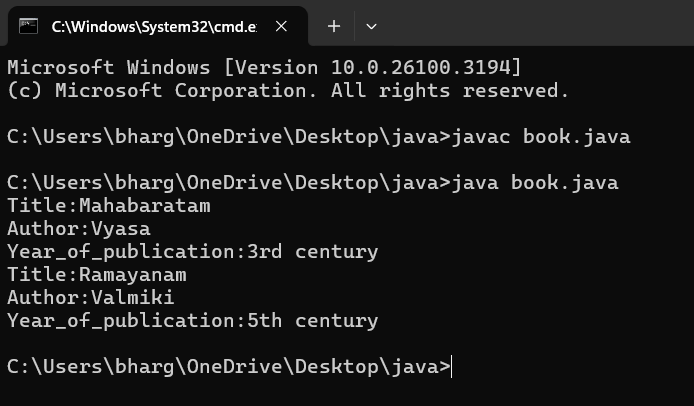
Book book2=new Book("Ramayanam","Valmiki","5th century");

book1.Bookdetails();

book2.Bookdetails();

}

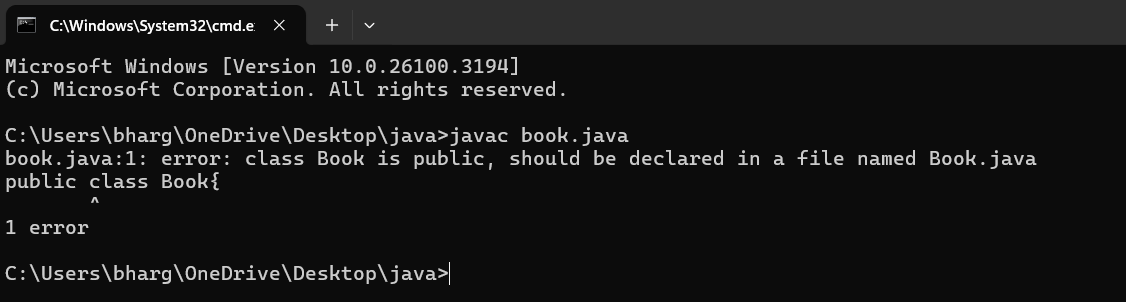
}



**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Not defining the function in a file. 2. Two public class files should not be saved in the same file. | 1. To call the method we must define a function in a file. 2. Two public class files should be saved in different files. |

**NEGATIVE CASE:**



**IMPORTANT POINTS:**

1. While defining two classes for a code, we must be sure that we save both the classes in separate files.
2. While defining a method we should also define a function to call that method.

**CLASS DIAGRAM:**

|  |
| --- |
| Book   * Title: String * Author: String * Year of publication: int   + Book(title: String,  Author: String;  Year of publication: int  + displayDetails( ): void |

**PROGRAM – 2:**

**AIM**: Create a java Program with class named myclass with static variable count of int type, initialized to zero and a constant variable “pi” of type double initialized to 3.14 as attributes of the class, ow define a constructor for “myclass” that increments the count variable each time an object of my class is created (count++), finally print the final values of count and pi variables create three objects.

**CODE:**

public class myclass {

static int count = 0;

final double pi = 3.14;

public myclass() {

count++;

}

public static void main(String[] args) {

myclass a = new myclass();

myclass b = new myclass();

myclass c = new myclass();

myclass d = new myclass();

myclass e = new myclass();

System.out.println("count: " + count);

System.out.println("Value of pi:"+a.pi);

System.out.println("Value of pi:"+b.pi);

System.out.println("Value of pi:"+c.pi);

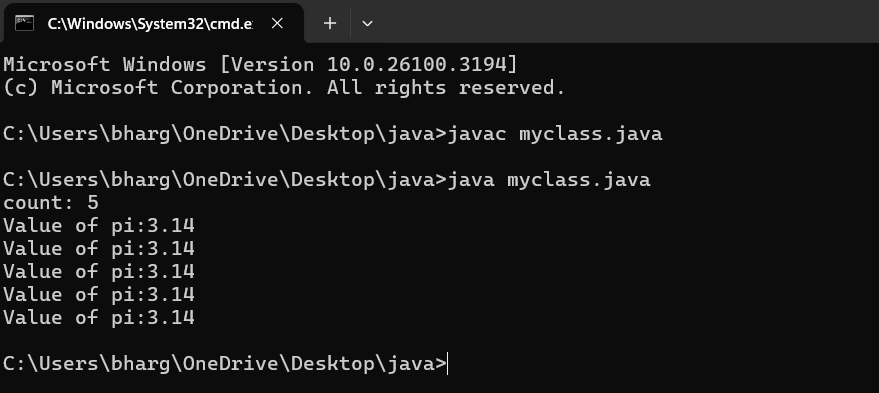
System.out.println("Value of pi:"+d.pi);

System.out.println("Value of pi:"+e.pi);

}

}

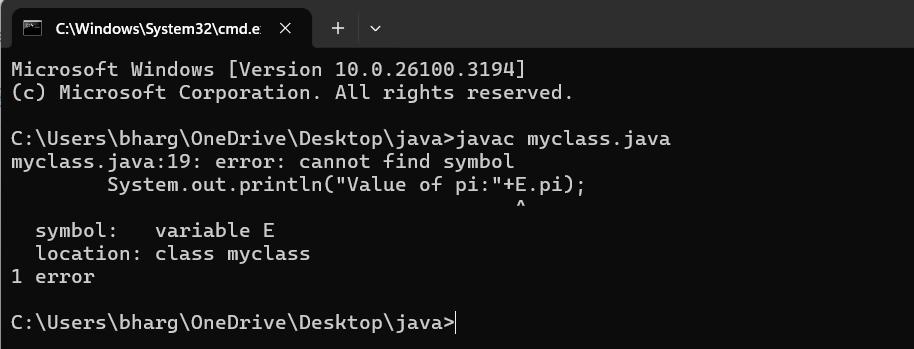
OUTPUT:



ERROR TABLE:

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Not Putting the semi-colon after calling a function, 2. Not giving the indentation properly. | 1. Put the semi-colon after calling a function. 2. All the indentation must be correct to run the code correct. |

**NEGATIVE CASE:**



**IMPORTANT POINTS:**

1. We must declare the initial value of the variable before declaring the final one.
2. Here the main objective is to increase the count according to the number of objects we make, i.e the count increases when the no.of objects are increasing.

**CLASS DIAGRAM:**

|  |
| --- |
| Myclass   * Count: int * Pi: double   + myclass( )  + main(args: String[]): void |