**23CSE111**

**OBJECT ORIENTED PROGRAMMING**

**LAB MANUAL**



ROLL NO :AV.SC.U4CSE235

NAME : N. JAYA RUDHRA MURTHY

SECTION : CSE-C

**1.AIM: To download**

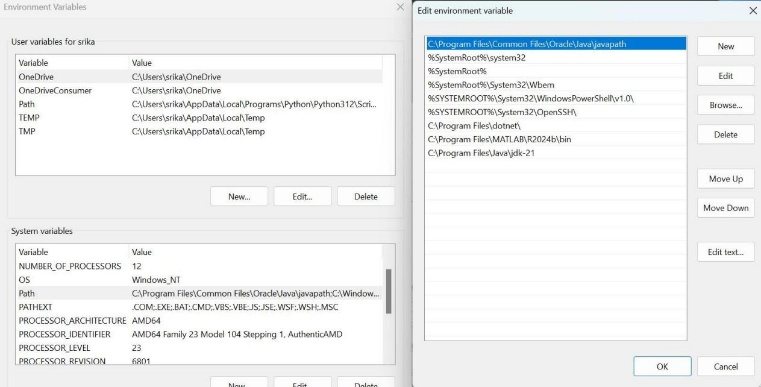
**Step – 1: Download JDK-21 from oracle website**



**Step-2: Install the JDK-21 with accepting terms and conditions according to the respective windows**

**Step-3: Setting up environmental variables***.*





1. Open Environment Variables:

Right-click This PC → Properties → Advanced system settings → Environment Variables.

2. Update JAVA\_HOME:

In System variables, find JAVA\_HOME.

Click Edit and set it to the path of the desired JDK (e.g., C:\Program Files\Java\jdk-21).

If it doesn’t exist, click New → Variable Name: JAVA\_HOME, Variable Value: C:\Program Files\Java\jdk-21.

3. Update Path:

Under System variables, select Path → Edit.

Add a new entry: %JAVA\_HOME%\bin.

3. Verify the Version

Open a new Command Prompt window and run:

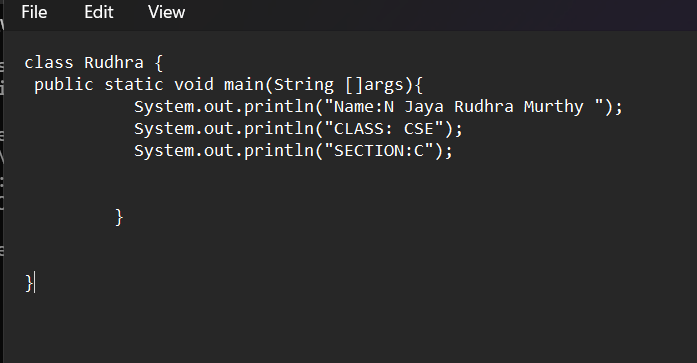
java -version

javac -version

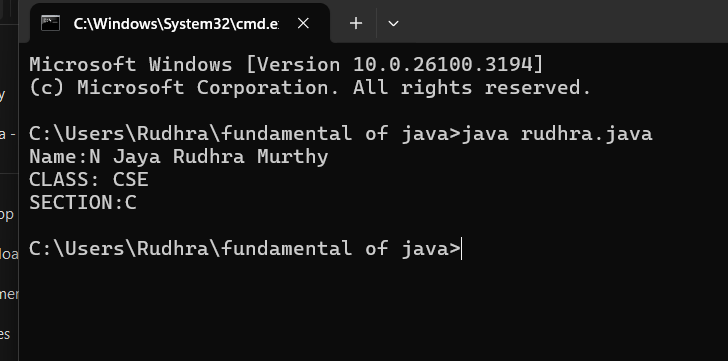
They should now display the updated Java version.

**1.AIM:**Simple Java Program for printing Name, Class, Roll No, of a Student

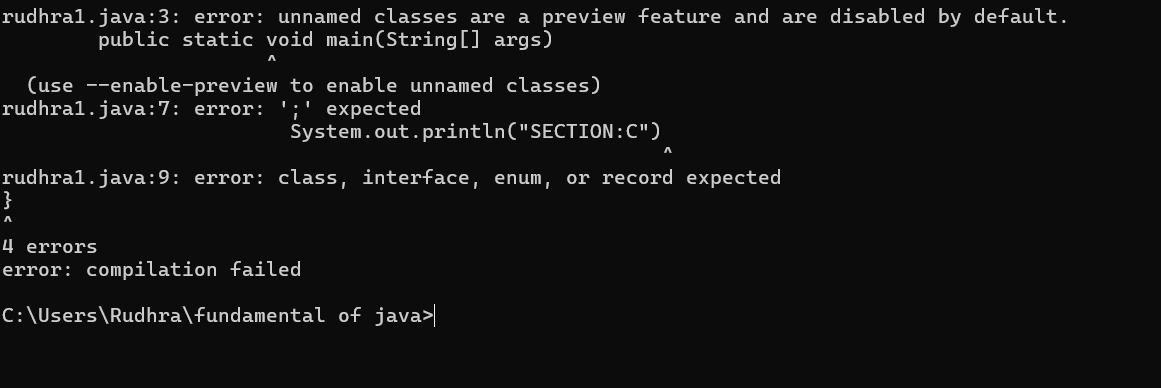
**CODE: -**

****

**Output: -**



Negative case:



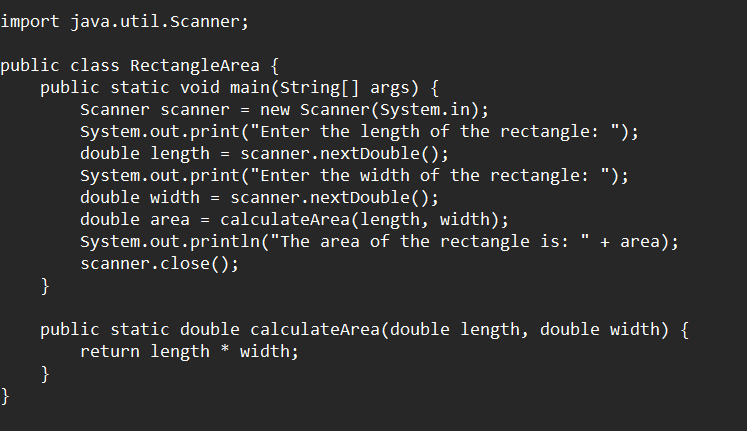
**Errors :**

|  |  |  |
| --- | --- | --- |
| **1** | **Syntax error** | **Semicolon added** |
| 2 | Runtime error | Copied correct path |
| 3 | Name error | rectified |

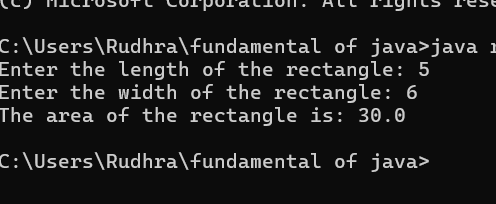
**Week -2 : SIMPLE JAVA PROGRAM**

1. **AIM:JAVA PROGRAM AREA OF RECTANGLE**

**Code:**

****

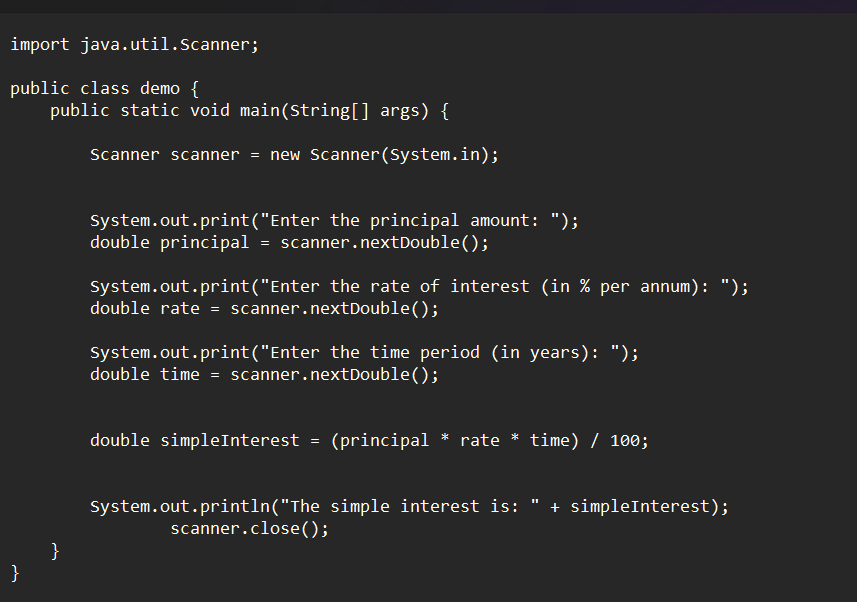
**Output:**

****

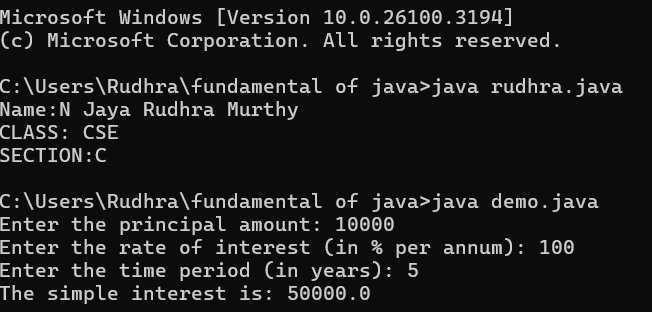
**Negative case:**

1. **AIM:Java program from calculate the simple interest**

**CODE:**

****

**Output:**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **rectification** |
| 1 | Runtime error | Incorrect path | Copied correct path |
| 2 | Syntax error | { missing | { added |
| 3 | Logical error | Wrong formula | Formula rectified |

1. **AIM: Java program for Fibonacci sieries**

**Syntax: import java.util.\*;**

**class fibo**

**{**

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

**{**

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

**int num;**

**int f3;**

**int f1 = 0;**

**int f2 = 1;**

**int i = 2;**

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

**num = sc.nextInt();**

**System.out.println(f1);**

**System.out.println(f2);**

**while(i<num)**

**{**

**}**

**f3 = f1+f2;**

**f1 = f2;**

**f2 = f3;**

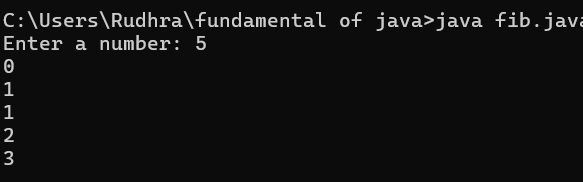
**System.out.println(f3);**

**i = i+1;**

**}**

**}**

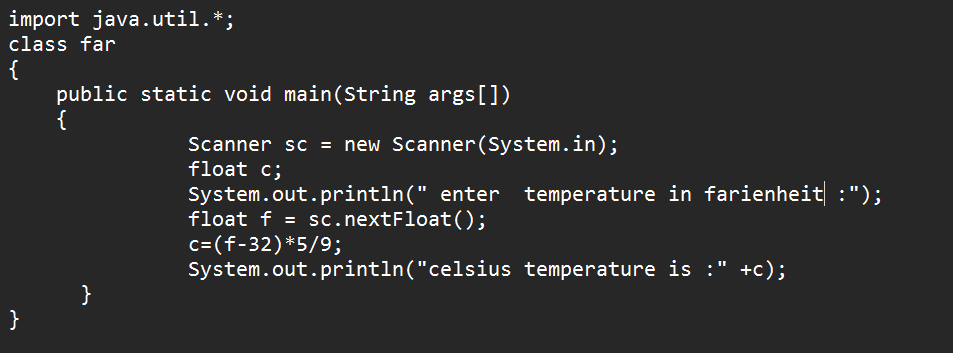
**Output:**



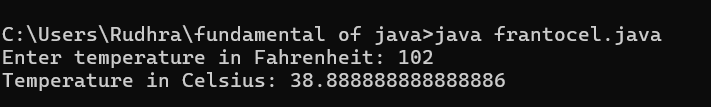
|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| **1** | **Name error** | **Incorrect usage of function** | **Correcting by using correct formula** |
| **2** | **Syntax error** | **No semicolon** | **Acolnidded sem** |
| **3** | **Runtime error** | **Incorrect path** | **Copied correct path** |

1. **AIM:Java program for calculating temperature in Celsius**

**CODE:**

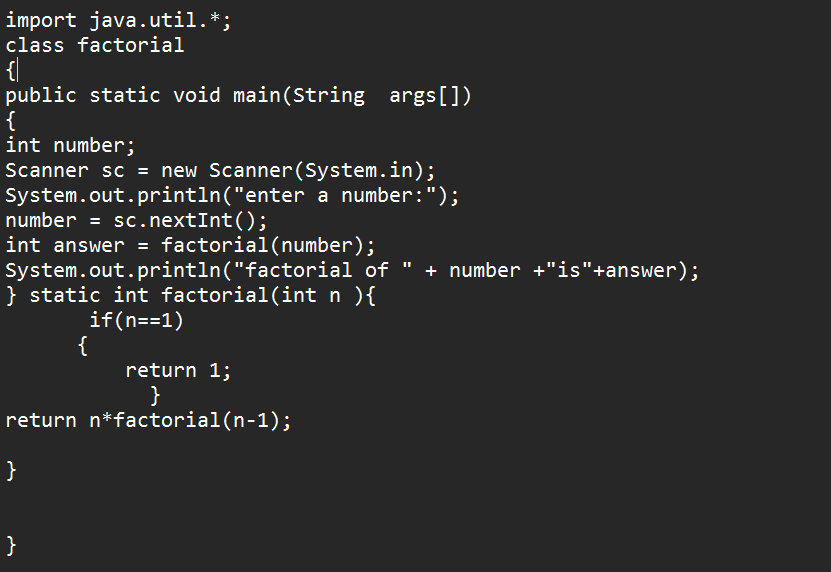
****

Output:

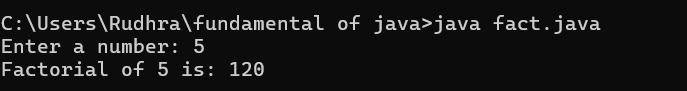


|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **rectification** |
| 1. | Syntax | No semicoln | Added semicoln |
| 2. | Logical error | Due to incorrect input | Corrected by giving correct input |
| 3. | Runtime error | Incorrect path | Using correct path |

**5.AIM:Java program factorial of a given number**

****

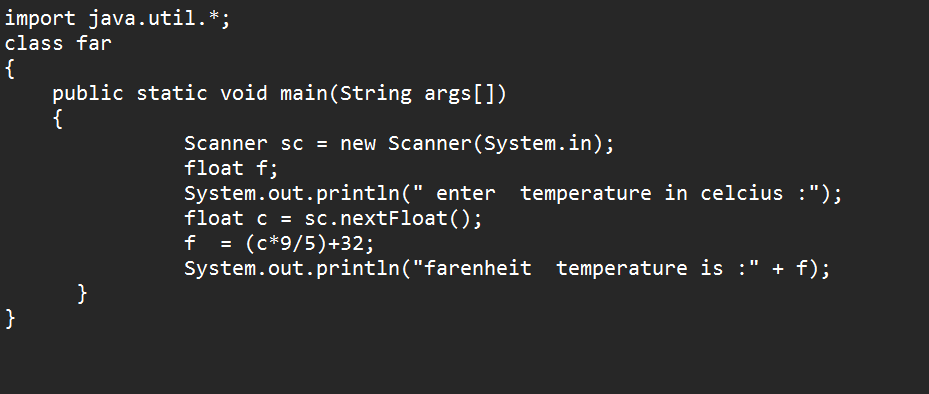
Output:



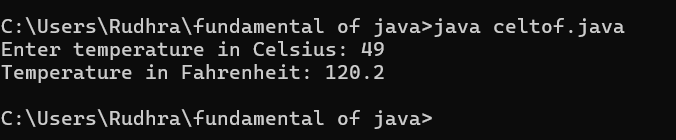
|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **rectification** |
|  | Logical error | Incorrect input | Correcting input |
|  | Runtime error | Incorrect path | Using correct path |
|  | syntax | No semicoln | Using semicoln |

**6.AIM:Java program for Celsius to farenheit:**

**CODE:**



Output:



**7.AIM:Write a program to find the area of triangle by using heron’s formula take the input from the user**

Code:



Output:



**Week -3:**

**1.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 three methods named start(), stop(). Service()**

**4. Create three objects named car1,car2 and car3**

**import java.util.\*;**

**class car**

**{**

**public String Car\_color;**

**public String Car\_brand;**

**public String fuel\_type;**

**public int mileage;**

**public void start()**

**{**

**System.out.println("Car Started:");**

**System.out.println("Car color is :"+Car\_color);**

**System.out.println("Car Brand is:"+Car\_brand);**

**System.out.println("Car fuel type is:"+fuel\_type);**

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

**}**

**public void service()**

**{**

**System.out.println("Car Started:");**

**System.out.println("Car color is :"+Car\_color);**

**System.out.println("Car Brand is:"+Car\_brand);**

**System.out.println("Car fuel type is:"+fuel\_type);**

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

**}**

**public void stop()**

**{**

**System.out.println("Car Started:");**

**System.out.println("Car color is :"+Car\_color);**

**System.out.println("Car Brand is:"+Car\_brand);**

**System.out.println("Car fuel type is:"+fuel\_type);**

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

**}**

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

**{ System.out.println("\nBHANU TEJA\n\n");**

**car car1 = new car();**

**car1.Car\_color = "Blue";**

**car1.Car\_brand = "Audi";**

**car1.fuel\_type = "Deisel";**

**car1.mileage = 100;**

**car1.start();**

**car car2 = new car();**

**car2.Car\_color = "Red";**

**car2.Car\_brand = "Tesla";**

**car2.fuel\_type = "EV";**

**car2.mileage = 200;**

**car2.stop();**

**car car3 = new car();**

**car3.Car\_color = "Yellow";**

**car3.Car\_brand = "BMW";**

**car3.fuel\_type = "Petrol";**

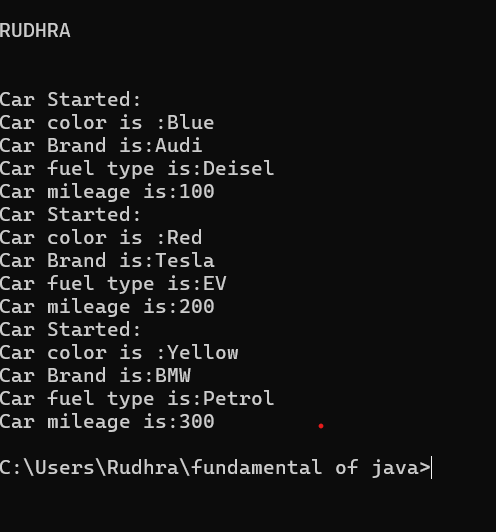
**car3.mileage = 300;**

**car3.service();**

**}**

**}**

**Output:**

****

**Class digram:**

|  |
| --- |
| car |
| - Car\_color: String  - Car\_brand: String  - fuel\_type: String  - mileage: int |
| + start(): void  + service(): void  + stop(): void |

Important points:

 The car class has four attributes: Car\_color, Car\_brand, fuel\_type, and mileage.

 It also has three methods: start(), service(), and stop().

 The start(), service(), and stop() methods all print the same details about the car.

 Each method prints the car's color, brand, fuel type, and mileage to the console

 The main method creates three instances of the car class: car1, car2, and car3.

 Each car object is assigned specific values for Car\_color, Car\_brand, fuel\_type, and mileage.

**2. AIM:To create a class bankAccount with methods deposit() and withdrawl**

**Code:**

**class BankAccount**

**{**

**private double balance;**

**public BankAccount(double initialBalance)**

**{**

**if(initialBalance > 0)**

**{**

**this.balance = initialBalance;**

**}**

**else**

**{**

**this.balance = 0;**

**}**

**}**

**public void deposit(double amount)**

**{**

**if(amount>0)**

**{**

**balance = balance+amount;**

**System.out.println("Deposited $:"+amount);**

**}**

**else**

**{**

**System.out.println("Deposited amount must be positive");**

**}**

**}**

**public double getBalance()**

**{**

**return balance;**

**}**

**}**

**public class Main1**

**{**

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

**{**

**BankAccount account = new BankAccount(1000);**

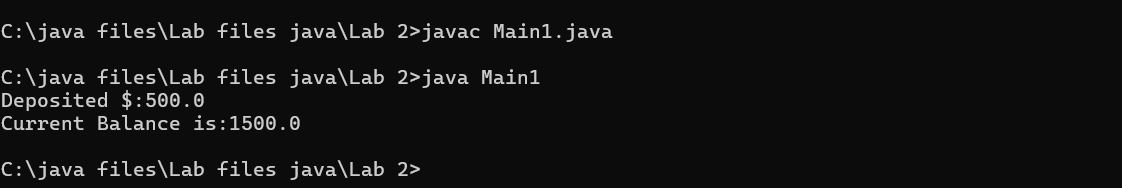
**account.deposit(500);**

**System.out.println("Current Balance is:"+account.getBalance());**

**}**

**}**

**Output:**



**Class digram:**

|  |
| --- |
| BankAccount |
| -balance:double |
| +BankAccount(doubleinitialBalance  +deposit(doubleamount): void  + getBalance(): double |

**Important points:**

The BankAccount class has a private attribute balance to store the account balance.

* The class has a constructor, BankAccount(double initialBalance), which initializes the balance. If the initial balance is not positive, it sets the balance to 0.
* The deposit(double amount) method adds a positive amount to the balance and prints a message. If the deposit amount is not positive, it prints an error message.
* The getBalance() method returns the current balance of the account.
* The Main1 class contains the main method, which serves as the entry point of the program.
* In the main method, an instance of BankAccount is created with an initial balance of 1000.

WEEK-4

**1.AIM:**

**WRITE A JAVA PROGRAM WITH CLASS NAMED “Book”. THE CLASS SHOUKD CONTAIN VARIOUS ATTRIBUTES SUCH AS TITLE, AUTHOR, YEAR OF**

**PUBLICATION. IT SHOULD ALSO CONTAIN A CONSTRUCTOR WITH**

**PARAMETERS WHICH INITIALIZES TITLE, AUTHOR, YEAR OF PUBLICATION**

**AND CREATE A METHOD WHICH DISPLAYS THE DETAILS OF 2 BOOKS.**

**PROGRAM:**

**class book{**

**public String title\_of\_book;**

**public String author;**

**public int year\_of\_publication;**

**public void start() {**

**System.out.println("Title of the book is :"+title\_of\_book);**

**System.out.println("Author of the book is :"+author);**

**System.out.println("Year of publication of the book is :"+year\_of\_publication);**

**}**

**public void service() {**

**System.out.println("Title of the book is :"+title\_of\_book);**

**System.out.println("Author of the book is :"+author);**

**System.out.println("Year of publication of the book is :"+year\_of\_publication);**

**}**

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

**book book1=new book();**

**book1.title\_of\_book="Harry Potter-The Goblet Of Fire";**

**book1.author="JK Rowling";**

**book1.year\_of\_publication=1992;**

**book1.start();**

**book book2=new book();**

**book2.title\_of\_book="Harry Potter-Deathy Hallows";**

**book2.author="JK Rowling";**

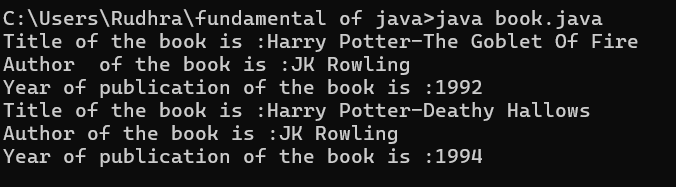
**book2.year\_of\_publication=1994;**

**book2.service();**

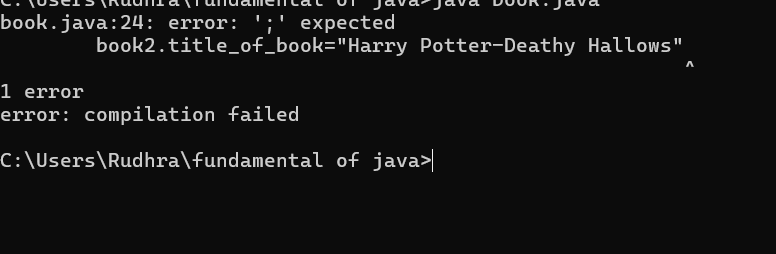
**}**

**}**

**Output:**

****

**Negative case:**



**ERROR:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **ERROR TYPE** | **Reason for error** | **Rectification** |
| **1.** | Syntax error | No semicolon | Semicolon added |
| **2.** | Runtime error | Incorrect path | Copied correct path |

**CLASS DIAGRAM:**

|  |
| --- |
| B  ook |
| -title: String  -author: String  -year: int |
| + Book(title: String, author:String, year: int) + displayDetails(): void |

**IMPORTANT POINTS:**

1. **Constructor**:

* The constructor Book(String, String, int) is used to initialize the object when it is created.
* The keyword **this** is used to differentiate between class attributes and constructor parameters.

2.**Method**:

* The method displayDetails() is used to display the book details.
* The **System.out.println()** method prints the details to the console.

3. **Object Creation**:

* Two objects b1 and b2 are created using the constructor.

**2.AIM:**

**WRITE A JAVA PROGRAM WITH CLASS NAMED “MyClass” WITH A STATIC VARIABLE COUNT OF INT TYPE. INTIALIZE IT TO ZERO AND A CONSTANT VARIABLE “Pi” OF TYPE DOUBLE INITIALIZED TO “3.14” AS ATTRIBUTES OF THAT CLASS. NOW DEFINE A CONSTRUCTOR FOR “MyClass”, THAT INCREMENTS THE COUNT VARIABLE EACH TIME AN OBJECT OF “MyClass” IS CREATED. FINALLY, PRINT THE FINAL VALUES OF ‘COUNT’ AND ‘PI’ VARIABLES AND CREATE 3 OBJECTS.**

**Program:**

**class myclass{**

**static int count = 0;**

**static final double pi = 3.14;**

**public myclass() {**

**count++;**

**}**

**public void start() {**

**System.out.println("Value of count : " + myclass.count);**

**System.out.println("Value of PI: " + myclass.pi);**

**}**

**public void service() {**

**System.out.println("Value of count : " + myclass.count);**

**System.out.println("Value of PI: " + myclass.pi);**

**}**

**public void stop(){**

**System.out.println("Value of count : " + myclass.count);**

**System.out.println("Value of PI: " + myclass.pi);**

**}**

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

**myclass obj1 = new myclass();**

**obj1.start();**

**myclass obj2 = new myclass();**

**obj2.service();**

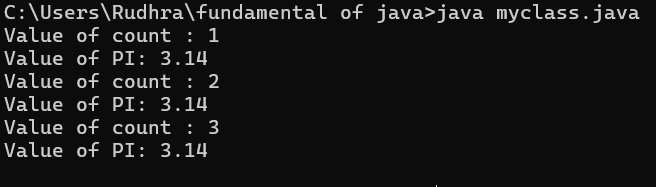
**myclass obj3 = new myclass();**

**obj3.stop();**

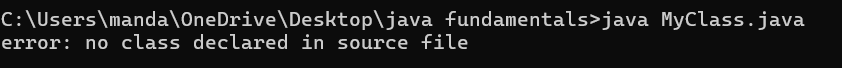
**}**

**}**

**Output:**

****

Negative case:



**ERROR:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error Type** | **Reason for error** | **Rectification** |
| **1.** | No class | No class name declared | Created class named ‘MyClass’ |
| **2.** | Syntax error | Not added keyword | Added keyword named ‘new’ |

**CLASS DIAGRAM:**

|  |
| --- |
| MyClass |
| -count: int (static)  -pi: double (static, final) |
| +MyClass()  +main(args: String[]):void |

**IMPORTANT POINTS:**

**1.Static Keyword**

* Static members belong to the **class, not to individual objects**.
* Only one copy of the static variable is maintained for all objects.

**2.Static Variable**

* **static int count**:
  + Shared among all objects of the class.
  + It is initialized only once and not for every object.
  + It increments every time the constructor is called.

**3.Final Variable**

* **static final double pi**:
  + The **final** keyword makes the variable constant.
  + Its value **cannot be changed** once assigned.
  + It must be initialized at the time of declaration.