OBJECT ORIENTED PROGRAMMING

LAB REPORT-23CSE111

VERIFIED BY:

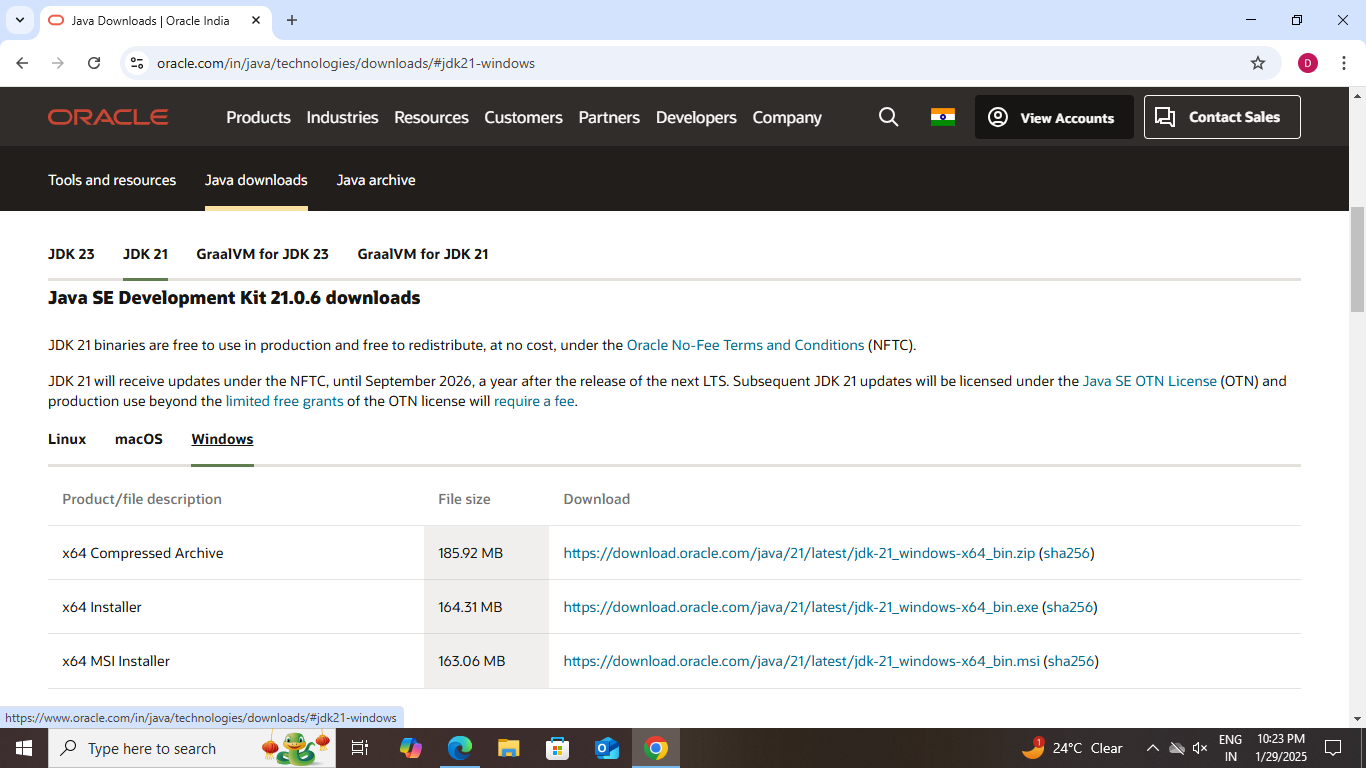
NAME:L.DILLESWARI

ROLLNO:24103

LAB-1:

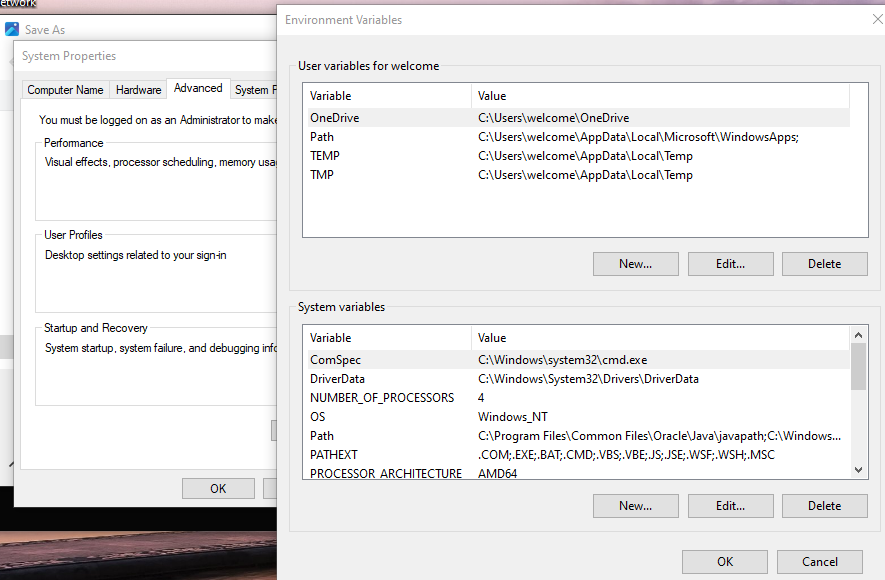
STEP TO DOWNLOAD AND INSTALL JDK:

1.Open web browser and download JDK21 X64 installer version as per your OS .



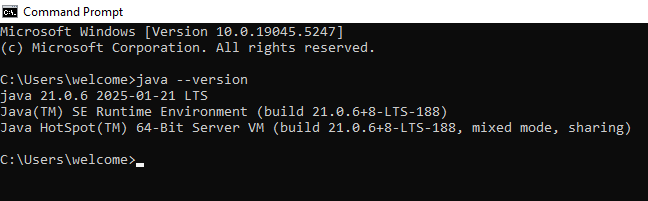
2.Run the installer until the installation is completed.

3.access the environment variable and click o the environment variable.



4. Find the path and edit and click new and it paste to your JDK bin path and click ok.

5.To verify your installation open  command prompt and use command (“java --version”).



**PROGRAM-1**

**AIM:** WRITE A PROGRAM TO PRINT DETAIL OF THE STUDENT

**CODE:**

Public class student {

    Public static void main(String[] args){

       System.out.println(“NAME : DILLESWARI **.** LOTTI”);

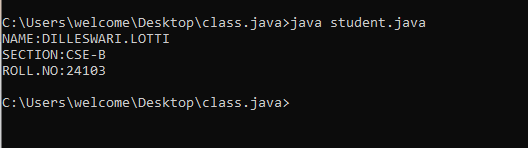
       System.out.println(“SECTION:CSE-B");

       System.out.println(“ROLLNO:24103”);

   }

}

**Output:**

****

**ERROR:**

|  |  |  |
| --- | --- | --- |
| **SLNO** | **ERROR** | **CORRECTION** |
| **1.** | error: ';' expected  System.out.println("ROLLNO:24103") | Insert: ‘;’  System.out.println(" ROLLNO:24103"); |
| **2.** | error: compilation error | Keep as need space |

**LAB-2**

**PROGRAM-1**

**AIM:** WRITE JAVA PROGRAM TO CALCULATE SIMPLE INTREST

**CODE:**

Import java.util.Scanner;

Public class si{

   Public Static void main(String[] args){

     Scanner input=new Scanner(System.in);

     System.out.println(“enter the principle:”);

    float P = input.nextFloat();

    System.out.println(“enter the time:”);

    int T = input.nextInt();

    System.out.println(“enter the rate:”);

     float R= input.nextFloat();

      Input.close();

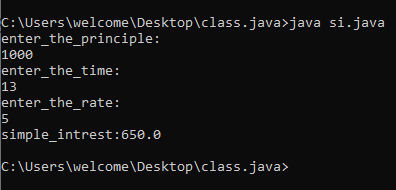
     Float si=(P\*T\*R)/100;

    System.out.println(“simple\_intrest:”+si);

  }

}

**OUTPUT:**

****

**ERROR**

|  |  |  |
| --- | --- | --- |
| **SLNO** | **ERROR** | **CORRECTION** |
| **1.** | error: ';' expected  System.out.print("Enter the rate of interest (R) in percentage: ") | Insert:‘;’  System.out.print("Enter the rate of interest (R) in percentage: "); |
| **2.** | error: cannot find symbol in Scanner scanner =new     scanner(System.in); | Replace capital S , Scanner scanner = new Scanner(System.in); |

**IMPORTANT POINTS:**

1.used Scanner library to get input from user in run time.

2.”importjava.util.Scanner;”-step to import library.

3.”Scanner input= new Scanner(System.in);”-step to use the scanner.[case sensitive]

4. Should give ‘;’ symbol at the end of System.out.print("Enter the rate of interest (R) in percentage: ").

**Program-2**

**Aim:**To write a program to change Fahrenheit to Celsius and vice versa

**Code:**

import java.util.Scanner;

class Fahrenheit {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the value of Celsius:");

        float C = scanner.nextFloat();

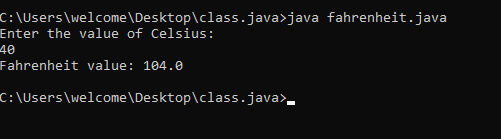
        float F = (C \* 9 / 5) + 32;

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

    }

}

**Output:**



**ERROR**

|  |  |  |
| --- | --- | --- |
| **SLNO** | **ERROR** | **CORRECTION** |
| **1.** | Scannner spelling mistake | Scanner |
| **2.** | Compilation error | Keep spaces as required |

**Important points:**

1. The System.out.println statement is using Celsius with a capital "C" which doesn't exist as a variable. Java is case-sensitive, so this will cause a compilation error. It should be lowercase celsius.

2. The scanner.close() method is called twice, which is unnecessary and could cause issues. It should only be called once.

**PROGRAM-3**

**AIM:** WRITE JAVA PROGRAM TO CALCULATE AREA OF RECTANGLE

**CODE:**

import java.util.Scanner;

public class rect{

   public static void main(String[] args){

Scanner input=new Scanner(System.in);

System.out.println("enter length");

double length=input.nextDouble();

System.out.println("enter width");

double width=input.nextDouble();

double area=length\*width;

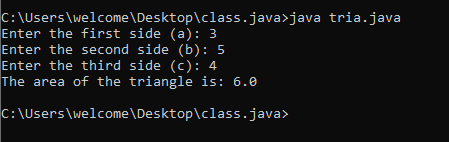
System.out.println("area="+area);

input.close();

    }

}

**Output:**



**ERROR:**

|  |  |  |
| --- | --- | --- |
| SLNO | ERROR | CORRECTION |
| 1. | int length = 5.0; // Should be double | **Replace:**double length |
| 2. | public class AreaOfRectangle {public static void main(String[] args) | **Replace:**  Add flower braces at theend. |

**Important points:**

1. Trying to assign a double value to an int variable.

2. Incorrect placement of braces,shouldbe placed correctly.

3.Should not give incorrect variables.

**PROGRAM-4**

**AIM:** WRITE JAVA PROGRAM TO CALCULATE AREA OF TRIANGLE USING HERONS FORMULA

**CODE:**

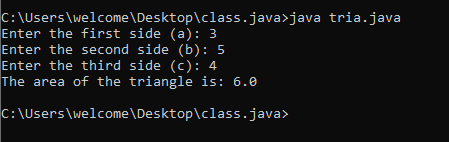
import java.util.Scanner;

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

   System.out.print("Enter the first side (a): ");   
    double a = input.nextDouble();   
    System.out.print("Enter the second side (b): ");   
    double b = input.nextDouble();   
    System.out.print("Enter the third side (c): ");   
    double c = input.nextDouble();    
    double s = (a + b + c) / 2;    
    double area = Math.sqrt(s \* (s - a) \* (s - b) \* (s - c));   
    System.out.println("The area of the triangle is: " + area);   
}

}

**Output:**



**ERRORS:**

|  |  |  |
| --- | --- | --- |
| SLNO | ERROR | CORRECTION |
| 1. | System “s” small | Replace it with System |
| 2. | ‘D’ small in  double c=input.nextdouble | double c =input.nextDouble |

**IMPORTANT POINTS:**

**1.**Verify that the input values can form a valid triangle (i.e., the sum of any two sides must be greater than the third side).

**2.**Calculate the semi-perimeter ss using the formula:

s=a+b+c/2.

**PROGRAM-5**

**AIM:** WRITE JAVA PROGRAM TO CALCULATE FIBONACI

**CODE:**

import java.util.Scanner;

public class Fib{

    public static void main(String[] args) {

        Scanner inp = new Scanner(System.in);

        // Input the number of terms to generate

        System.out.print("Enter the number of terms in the Fibonacci sequence: ");

        int n = inp.nextInt();

        // Handle edge cases

        if (n <= 0) {

            System.out.println("Please enter a positive integer.");

        }

else if (n == 1) {

            System.out.println("Fibonacci sequence: 0");

        }

else {

            System.out.println("Fibonacci sequence:");

            int first = 0, second = 1;

            // Print the first two terms

            System.out.print(first + ", " + second);

            // Generate the remaining terms

            for (int i = 3; i <= n; i++) {

                int next = first + second;

                System.out.print(", " + next);

                first = second;

                second = next;

            }

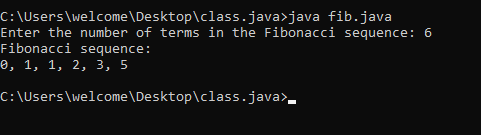
            System.out.println(); // Newline after the sequence

         }

     }

}

OUTPUT:



|  |  |  |
| --- | --- | --- |
| SLNO | ERROR | CORRECTION |
| 1. | int firstTerm;  Error: variables not initialized properly | should be initialized          int n = 10; |
| 2. | Int[] fibonacci = new int[n];  System.out.println(fibonacci[n + 1]); | invalid access  Replace:System.out.println("Fibonacci Series up to " + n + " terms:"); |

**IMPORTANT POINTS:**

1. Variables should be initialized properly
2. Accessing an array element outside its bounds.
3. Loop condition should be properly defined. No syntax error should be there.

LAB-3

1. **To create java program with following :**
2. **Create a class with name car**
3. **Create four attributes named Car\_colour,Car\_brand,Fuel\_type,mileage**
4. **Create three methods named Strat(),Stop(),Service()**
5. **Create three obj named Car1,Car2,Car3.**

**CODE FOR CREATING CAR CLASS**

class car

{

public String car\_color;

public String car\_brand;

public String fuel\_type;

public float mileage;

public void start()

{

System.out.println("Car starts");

}

public void stop()

{

System.out.println("Car stops");

}

public void service()

{

System.out.println("Car service");

}

public static void main(String [] args){

// object one creation

car car1= new car();

car1.car\_color="Red";

car1.car\_brand="Ferrari";

car1.fuel\_type="Petrol";

car1.mileage=75.69F;

//calling methods for object 1

car1.start();

car1.stop();

car1.service();

System.out.println("color of the car1 is "+car1.car\_color);

System.out.println("brand of the car1 is "+car1.car\_brand);

System.out.println("fuel type of the car1 is"+car1.fuel\_type);

System.out.println("mileage of the car1 is"+car1.mileage);

// object two creation

car car2= new car();

car2.car\_color="Blue";

car2.car\_brand="BMW";

car2.fuel\_type="Petrol";

car2.mileage=11.96F;

// calling methods for object 2

car2.start();

car2.stop();

car2.service();

System.out.println("color of the car2 is "+car2.car\_color);

System.out.println("brand of the car2 is "+car2.car\_brand);

System.out.println("fuel type of the car2 is "+car2.fuel\_type);

System.out.println("mileage of the car2 is"+car2.mileage);

//object three creation

car car3= new car();

car3.car\_color="black";

car3.car\_brand="Audi";

car3.fuel\_type="Diesel";

car3.mileage=11.76F ;

// calling methods for object 3

car3.start();

car3.stop();

car3.service();

System.out.println("color of the car3 is "+car3.car\_color);

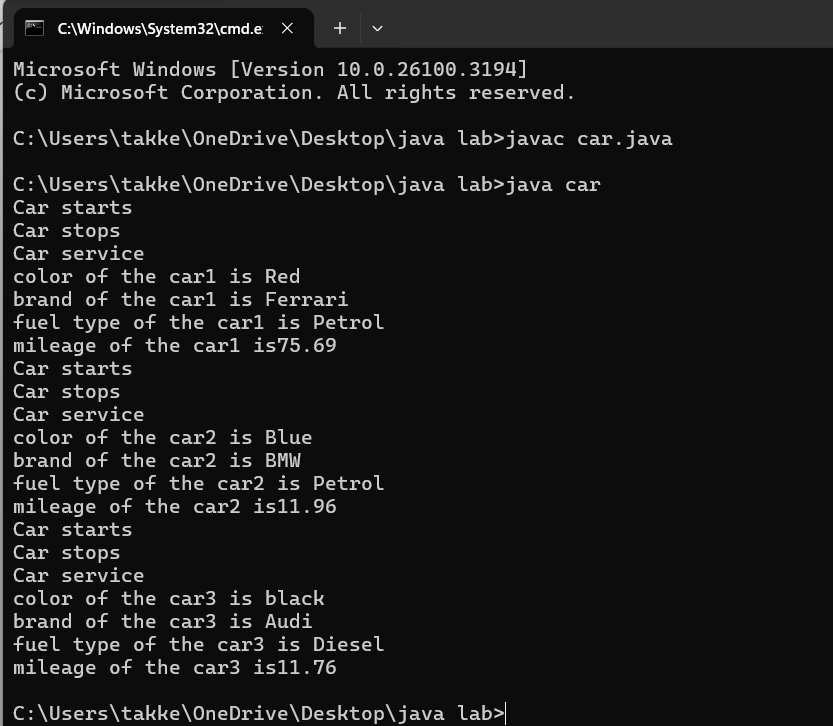
System.out.println("brand of the car3 is "+car3.car\_brand);

System.out.println("fuel type of the car3 is "+car3.fuel\_type);

System.out.println("mileage of the car3 is"+car3.mileage);

}

}

**output**

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| **SLNO** | **ERROR** | **CORRECTION** |
| **1.** | Error: car\_Color is undefined, should be car\_color | Replace:car\_Color with car\_color |
| **2.** | Missing semicolon after 'car2.start()' will cause an error | Replace: add semi colon at the end |

**IMPORTANT POINTS:**

**1. Variable name mismatch**: The variable car\_Color in the code should be car\_color

2.**Incorrect variable name**: car1.car\_color is used when the actual variable is car1.car\_Color, which will cause an error due to case sensitivity.

3. **Missing Semicolon**: Forgetting to add a semicolon at the end of a statement will cause a compilation error.

**2.To create a class Bank Account with Methods deposit() and Withdrawal()**

**CODE:**

import java.util.Scanner;

class BankAccount {

private float existing; // Class-level variable to store balance

private Scanner input; // Single Scanner instance for input

// Constructor

public BankAccount() {

input = new Scanner(System.in);

System.out.print("Enter existing amount in bank account: ");

this.existing = input.nextFloat();

}

// Deposit method

public void deposit() {

System.out.print("Enter amount to be deposited: ");

float deposit = input.nextFloat();

existing += deposit;

System.out.println("Existing amount now is: " + existing);

}

// Withdrawal method

public void withdrawal() {

System.out.print("Enter amount to be withdrawn: ");

float withdrawal = input.nextFloat();

if (existing < withdrawal) {

System.out.println("Not sufficient balance.");

} else {

existing -= withdrawal;

System.out.println("Remaining balance: " + existing);

}

}

// Main method

public static void main(String[] args) {

BankAccount customer1 = new BankAccount();

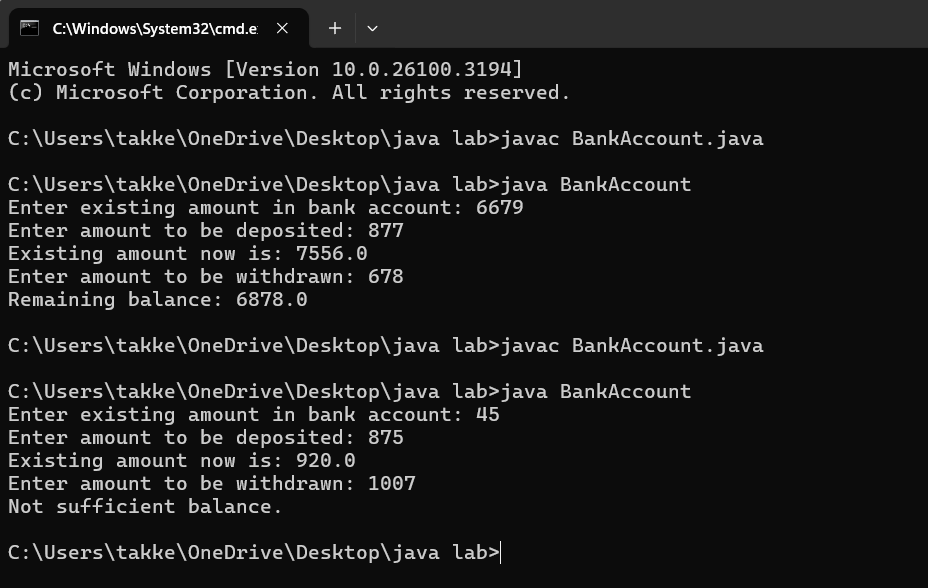
customer1.deposit();

customer1.withdrawal();

}

}

**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| **SI.NO** | **ERROR** | **correction** |
| **1.** | this.existing = int.nextFloat(); | this.existing = input.nextFloat(); |
| **2.** | public int deposit() is incorrect return method | REPLACE:public VOID deposit() |

**IMPORTANT POINTS:**

1.The balance should be a float or double to handle decimal values correctly, but it's declared as an int.

2. **Incorrect deposit method signature**: The method DEPOSIT()has an incorrect return type int(), while it should be void since it doesn't need to return any value.

3. **Fixed the return type of deposit**: Changed from int to void, as the method does not need to return anything

**LAB-4**

**AIM :: Write a java program with class named Book**

**a)a class should contain various attributes such as title,author,year of publication.**

**b)it should also contain a constructor with parameters which initializes title,author,year of publication.**

**c)create a method which displays the details of the book title ,author ,year of publication**

**Display the details of two books.**

**CODE:**

class Book {

// Fields to hold book details

public String bookTitle;

public String bookAuthor;

public int bookYearOfPublication;

// Method to print book title

public void title() {

System.out.println("Book Title");

}

// Method to print book author

public void author() {

System.out.println("Book Year of Publishing");

}

// Main Method

public static void main(String[] args) {

// First book

Book book1 = new Book();

book1.bookTitle = "Atomic Habits";

book1.bookAuthor = "James Clear";

book1.bookYearOfPublication = 2018;

// Calling methods

book1.title();

book1.author();

// Printing book details

System.out.println("Book title is: " + book1.bookTitle);

System.out.println("Book author is: " + book1.bookAuthor);

System.out.println("Book year of publication is: " + book1.bookYearOfPublication);

// Second book

Book book2 = new Book();

book2.bookTitle = "South Pole Pig";

book2.bookAuthor = "James";

book2.bookYearOfPublication = 2014;

// Calling methods

book2.title();

book2.author();

// Printing book details

System.out.println("Book title is: " + book2.bookTitle);

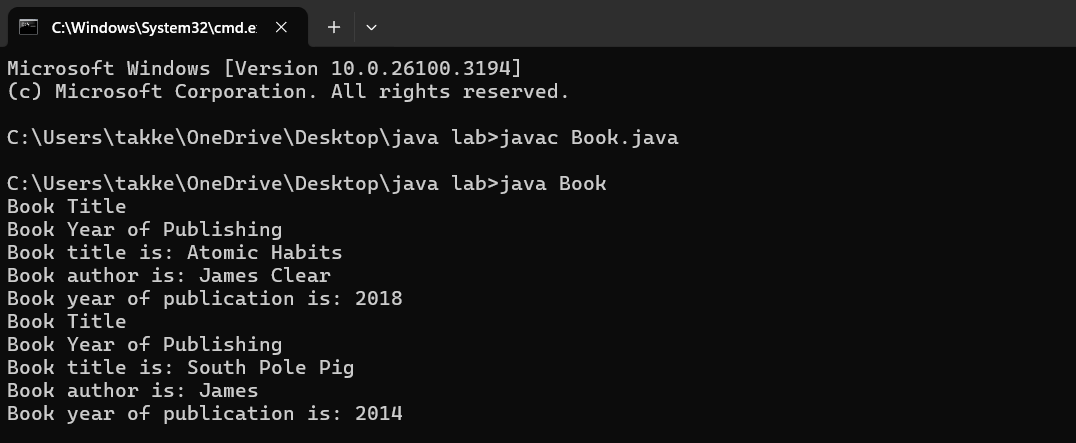
System.out.println("Book author is: " + book2.bookAuthor);

System.out.println("Book year of publication is: " + book2.bookYearOfPublication);

}

}

OUTPUT:

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| **SI.NO** | **ERROR** | **CORRECTION** |
| **1.** | Not defining the function in a file. | To call the method we must define a function in a file. |
| **2.** | Two public class files should not be saved in the same file. | Two public class files should be saved in different files. |

**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 |

**2. Create a java program with class named ‘MyClass’ with static variable count of int type,initaialized to zero and a constant variable ‘pi’ or type double initialized to 3.14 as attributes of the class. Now define a constructor for “MyClass”that increments the count variable eachtime an object of MyClass is created.Finally print the final values of count and pi variables.**

**Create three objects and a constructor.**

**CODE:**

class MyClass {

// Static variable to hold count of objects created

static int count = 0;

// Constant variable for pi, initialized to 3.14

final double pi = 3.14;

// Constructor

public MyClass() {

// Increment count every time an object is created

count++;

}

// Main method to create objects and print values

public static void main(String[] args) {

// Creating three objects of MyClass

MyClass object1 = new MyClass();

MyClass object2 = new MyClass();

MyClass object3 = new MyClass();

// Printing the final values of count and pi

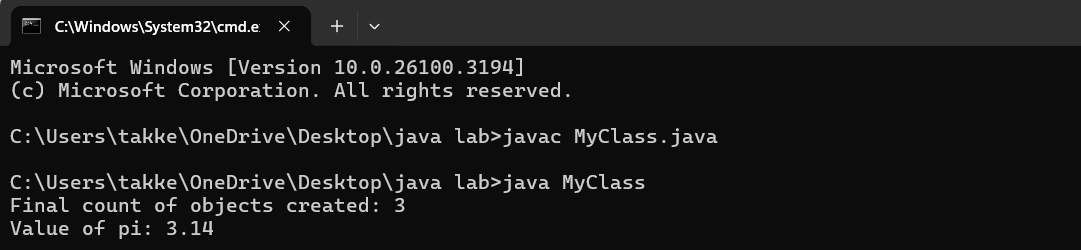
System.out.println("Final count of objects created: " + count);

System.out.println("Value of pi: " + object1.pi); // Pi is constant, and we can access it via any object

}

}

OUTPUT:

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| **SI.NO** | **ERROR** | **CORRECTIONS** |
| **1.** | Not Putting the semi-colon after calling a function, | Put the semi-colon after calling a function. |
| **2.** | Not giving the indentation properly. | All the indentation must be correct to run the code correct |

**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 |