**23CSE111**

**OBJECT-ORIENTED PROGRAMMING**

**LAB REPORT**

****

**Department of Computer Science Engineering**

**Amrita School of Computing**

**Amrita Vishwa Vidyapeetham, Amaravati Campus**

**VERIFIED BY: NAME: K. MANASA**

**ROLL NO: AV.SC.U4CSE24206**

**WEEK 01**

**PROGRAM-1:**

**AIM:** Download and Install Java Software

**PROCEDURE:**

**Step 1: Download JDK 21**

1. Open your web browser and go to the Oracle JDK Downloads page
2. Scroll down to the Java SE Development Kit 21 section.
3. Choose the Windows x64 Installer version.
4. Click on Download, then Wait for the download to complete**.**



**Step 2:** **Install JDK 21**

1. Locate the downloaded jdk-21\_windows-x64\_bin.exe file.
2. Double-click to launch the installer.
3. Click Next on the setup wizard.
4. Choose the installation path (default is C:\Program Files\Java\jdk-21).
5. Click Next, then click Install.
6. Wait for the installation to complete.
7. Click Close once the installation is finished.



**Step 3: Setting up the path**

1) Go to “Windows C” Drive on Desktop

2) Choose Program Files, select Java, then JDK 21, then select Bin.

 3) Select and copy the path at the address bar.

**Step 4: Open System Properties**

1. Press Windows + R, type sysdm.cpl, and click Ok-
2. The System Properties window will open.
3. Navigate to the Advanced tab.
4. Click on Environment Variables at the bottom.

**Step 5: Set JAVA\_HOME**

1)Under System Variables, click New.

2)Set the Variable name as JAVA\_HOME.

3)Set Variable value as C:\Program Files\Java\jdk-21 (or your installation path).

4)Click OK.



**Step 6: Update PATH Variable**

1)In System Variables, find Path and click Edit.

2)Click New and add: C:\Program Files\Java\jdk-21\bin

3)Click OK to save.



**Step 7:Verify Installation**

1. Open Command Prompt.
2. Type the following command: **java --version** and press Enter.



**PROGRAM-2:**

**AIM:** Write a Java program to print the message Hello World

”

**CODE:**

Public class helloworld {

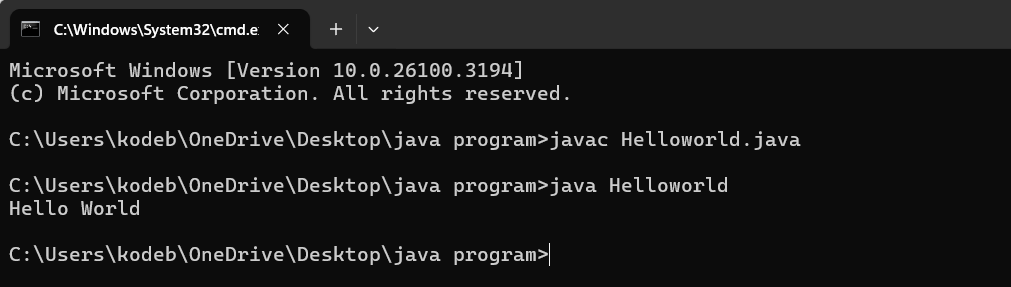
public static void main (String [] args) {

System. out. println ("Helloworld");

}

}

**OUTPUT:**



**ERRORS:**

|  |  |
| --- | --- |
| Error found | Error rectified |
| The class name should be in capital letter | Mention the first letter of the class name by capital letter |

**PROGRAM-3:**

**AIM:** Write a Java Program that prints Name, Roll No, Section of a student.

**CODE:**

class studentdetails{

public static void main (String [] args) {

System.out.println("Name: K.Manasa");

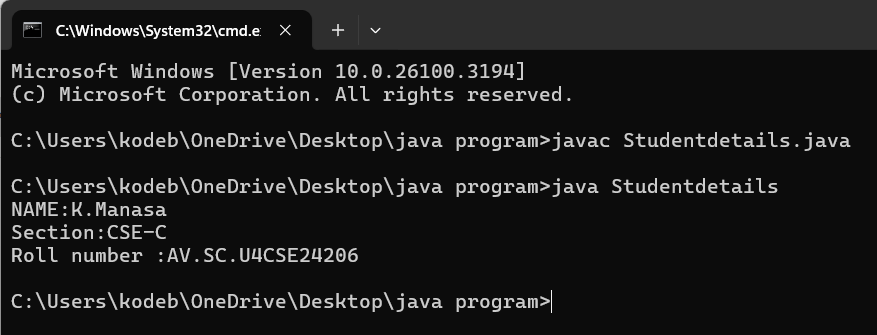
System.out.println("Section: CSE-C");

System.out.println("Roll no:AV.SC.U4CSE24206");

}

}

**OUTPUT:**



**ERRORS:**

|  |  |
| --- | --- |
| Error found | Error rectified |
| cannot find symbol  public static void main(string args[]) {  ^ | public static void main (String [] args) { |

**WEEK 02**

**PROGRAM-1:**

**AIM:** Write a Java program to calculate area of rectangle and area of triangle using herons formula.

**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();

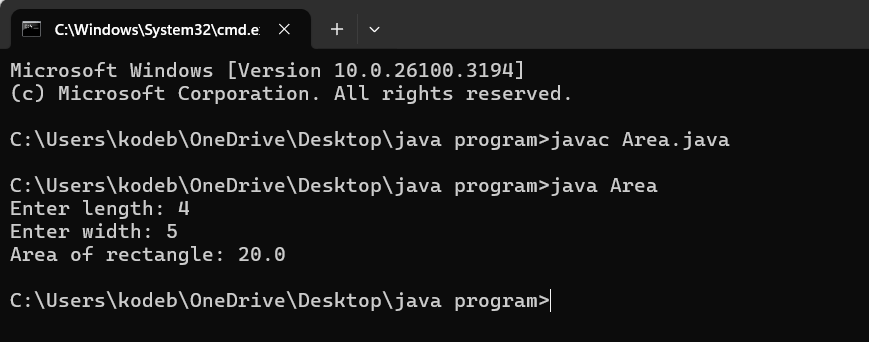
float area = l \* b;

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

}

}

**OUTPUT:**



**ERRORS:**

|  |  |
| --- | --- |
| Error found | Error rectified |
| Area.java:7: error: cannot find symbol  float l = inputnextFloat();  ^  symbol: method inputnextFloat()  location: class Area | float l = input.nextFloat(); |

**CODE:**

public class Areat {

public static void main(String[] args) {

double s1, s2, s3;

double area, resArea;

s1 = 35.0;

s2 = 8.0;

s3 = 38.0;

area = (s1+s2+s3)/2.0d;

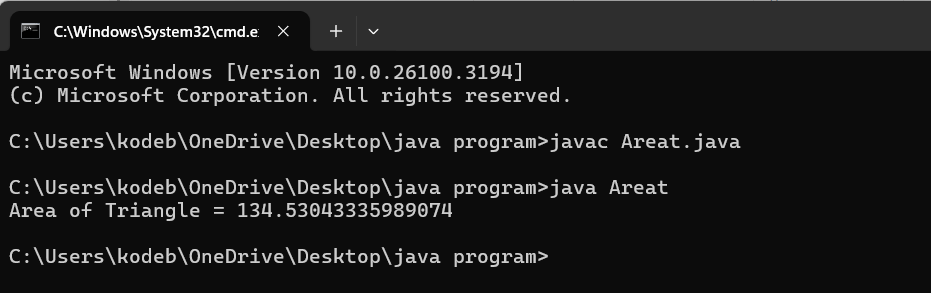
resArea = Math.sqrt(area\* (area - s1) \* (area - s2) \* (area - s3));

System.out.println("Area of Triangle = " + resArea);

}

}

**OUTPUT:**

****

**ERRORS:**

|  |  |
| --- | --- |
| Error found | Error rectified |
| Areat.java:5: error: ';' expected  double area resArea;  ^  Areat.java:5: error: not a statement  double area resArea;  ^ | double area, resArea; |

**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();

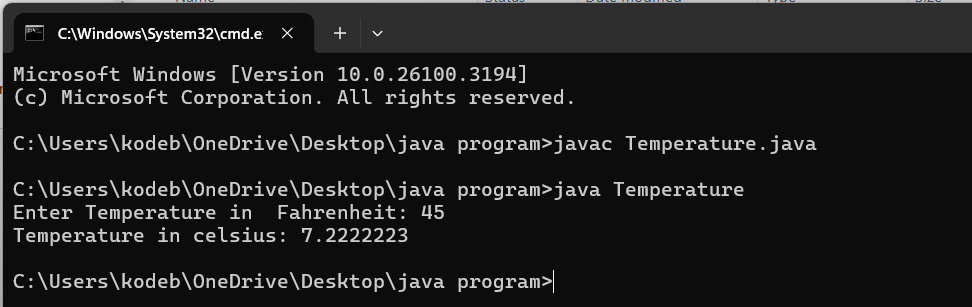
float C = (F - 32)\*5/9;

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

}

}

**OUTPUT:**



**ERRORS:**

|  |  |
| --- | --- |
| Error found | Error rectified |
| S should be capital in System.out.print | System.out.println("Temperature in celsius: " + C); |

**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();

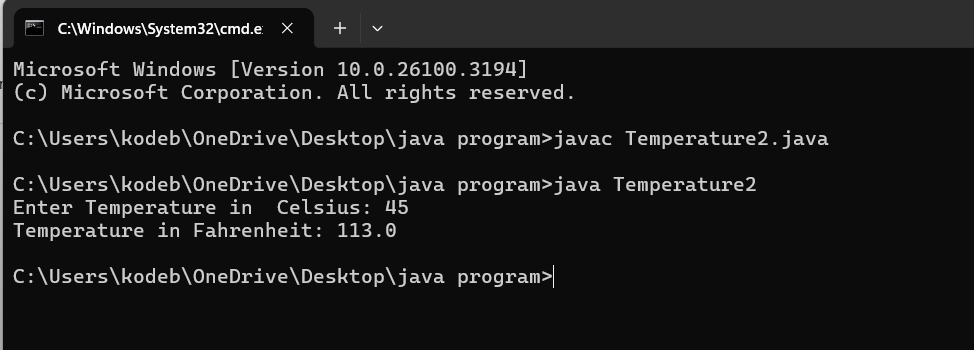
float F = (C \* 9/5) + 32;

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

}

}

**OUTPUT:**

****

|  |  |
| --- | --- |
| Error found | Error rectified |
| Temperature2.java:13: error: package system does not exist  system.out.println("Temperature in Fahrenheit: " + F); | System.out.println("Temperature in Fahrenheit: " + F); |

**PROGRAM-3:**

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

**CODE:**

import java.util.Scanner;

class Simpleinterest {

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();

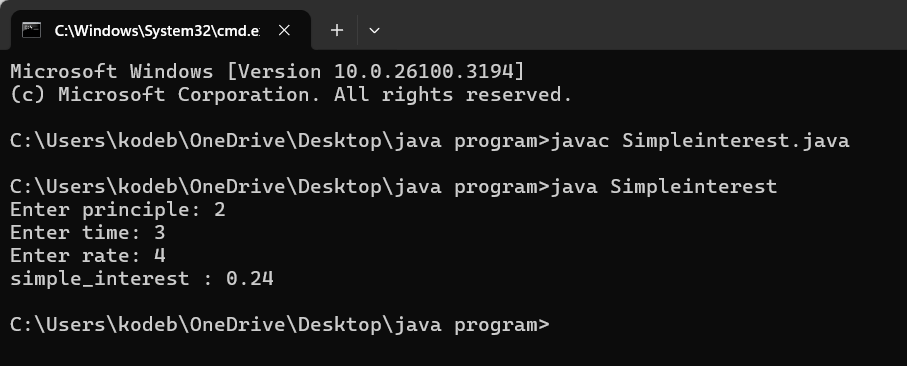
float SI = (P\*T\*R)/100;

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

}

}

**OUTPUT:**

****

**ERROR:**

|  |  |
| --- | --- |
| Error found | Error rectified |
| unclosed string literal  System.out.print("Enter time: );  ^ | System.out.print("Enter time: "); |

**PROGRAM-4:**

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

**CODE:**

import java.util.Scanner;

public 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;

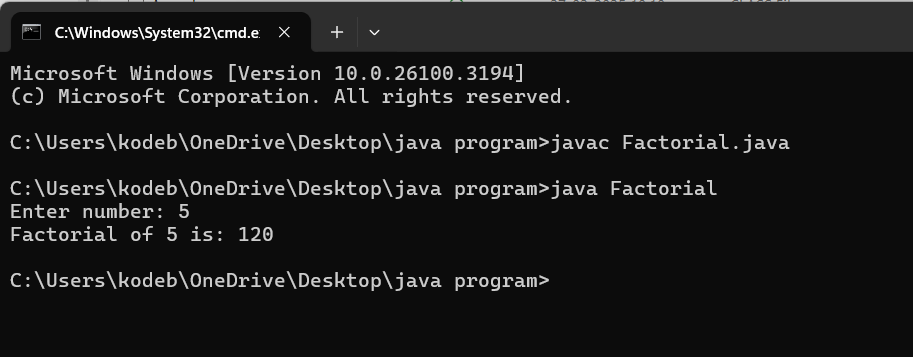
}

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

}

}

**OUTPUT:**



**ERROR:**

|  |  |
| --- | --- |
| Error Found | Error rectified |
| Factorial.java:2: error: class, interface, enum, or record expected  Public class Factorial {  ^ | public class Factorial { |

**PROGRAM-5**

**AIM:** To Write a java program on Fibonacci series

**Code:**

import java.util.Scanner;

public class Fibonacci {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int terms = scanner.nextInt();

long firstTerm = 0, secondTerm = 1;

System.out.println("Fibonacci Series up to " + terms + " terms:");

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

System.out.print(firstTerm + " ");

long nextTerm = firstTerm + secondTerm;

firstTerm = secondTerm;

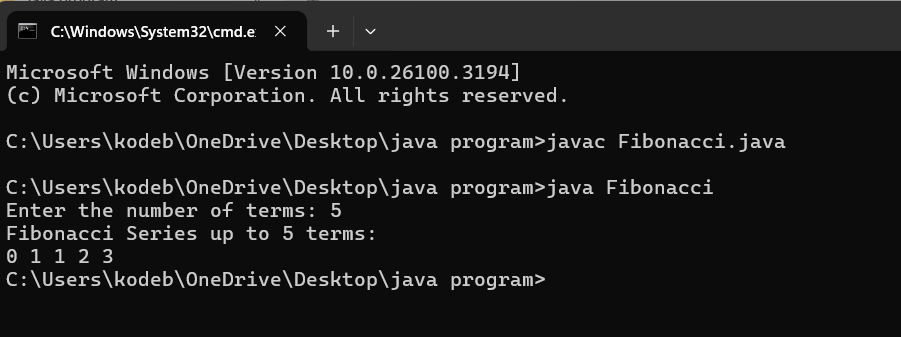
secondTerm = nextTerm;

}

}

}

**OUTPUT:**

****

**ERRORS:**

|  |  |
| --- | --- |
| Error found | Error rectified |
| Fibonacci.java:7: error: ';' expected  int terms = scanner.nextInt()  ^ | At the end int terms = scanner.nextInt() this line we should keep ; |

**WEEK 03**

**PROGRAM-1:**

**AIM:** Write a java program with  
 1. A class with name Car.  
 2.Create 4 attributes named car\_color , car\_brand ,fuel\_type ,mileage.

3.Create 3 methods named start() ,stop() ,service().  
 4.Create 3 objects named car1 ,car2 ,car3.  
**CODE:**class Car {  
 private String car\_color;  
 private String car\_brand;  
 private String fuel\_type;  
 private double mileage;  
 public Car(String car\_color,String car\_brand,String fuel\_type,double mileage){  
 this.car\_color = car\_color;  
 this.car\_brand = car\_brand;  
 this.fuel\_type = fuel\_type;  
 this.mileage = mileage;  
 System.out.println("the car is described as "+car\_color+","+car\_brand+","+fuel\_type+","+mileage+" miles");   
 }  
 public void start(String car\_brand){  
 System.out.println(car\_brand +" is starting");  
 }  
 public void stop(String car\_brand){  
 System.out.println(car\_brand +" is stoping");  
 }  
 public void service(String car\_brand){  
 System.out.println(car\_brand +" needs service");  
 }  
 public static void main(String[] args){  
 // object 1 is taken as car1  
 Car car1 = new Car("White","Benz","Diesel",47.9);  
 car1.start("Benz");  
 car1.stop("Benz");  
 car1.service("Benz");  
 // object 2 is taken as car 2  
 Car car2 = new Car("Red","Ferrari","Petrol",14.6);  
 car2.start("Ferrari");  
 car2.stop("Ferrari");  
 car2.service("Ferrari");  
 // object 3 is taken as car 3  
 Car car3 = new Car("Grey","Lamborghini","Gasoline",22);  
 car3.start("Lamborghini");  
 car3.stop("Lamborghini");  
 car3.service("Lamborghini");  
 }  
}

**CLASS DIAGRAM:**

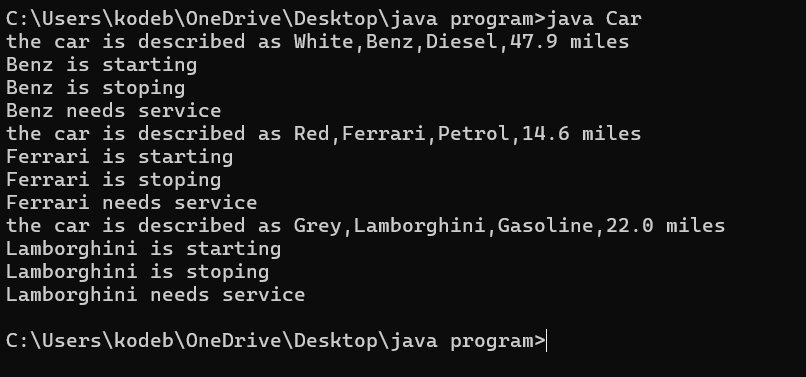
Class Car

- car\_color : String  
- car\_brand : String  
- fuel\_type : String  
- mileage : Double

+ start() : void  
+ stop() : void  
+ service(): void

**IMPORTANT POINT :** 1. The Car class defines attributes such as car\_color, car\_brand, fuel\_type, and mileage, all of which are private fields**.** 2. The constructor public Car(String car\_color, String car\_brand, String fuel\_type, double mileage) initializes the attributes of the Car object and prints a description of the car. A constructor means having the same name as class.3. The Car class includes three methods: start(String car\_brand), stop(String car\_brand), and service(String car\_brand).

**OUTPUT:**

****

**ERRORS:**

|  |  |
| --- | --- |
| Errors Found | Errors Rectification |
| I typed private car\_color; | I rectified it as private String car\_color; |
| I wrote it as System.out.println(+car" is starting"); | I rectified it as System.out.println(car\_brand+"is starting"); |

**PROGRAM 2 :  
AIM :** Write a java program with  
 1.Create a class named Bankaccount.  
 2.Create a constructor.  
 3.Create 2 methods which are withdrawl() and deposit().  
**CODE:**class Bankaccount {  
 private String AccName;  
 private int AccNo;  
 private float currentbalance;  
 public Bankaccount(String AccName,int AccNo,float currentbalance){  
 this.AccName=AccName;  
 this.AccNo=AccNo;  
 this.currentbalance=currentbalance;  
 System.out.println("the customer details are "+AccName+","+AccNo+","+currentbalance);  
 }  
 public void withdraw(int amount){  
 if(amount< currentbalance){  
 currentbalance=currentbalance-amount; System.out.println("the currentbalance is "+currentbalance);   
 }  
 }  
 public void deposit(int amount){ currentbalance=currentbalance+amount; System.out.println("the currentbalance is "+currentbalance);  
 }   
 public static void main(String[] args){  
 Bankaccount p1 = new Bankaccount("Manasa",12072007,10000000); p1.withdraw(700000);  
 p1.deposit(500000);  
 }  
}

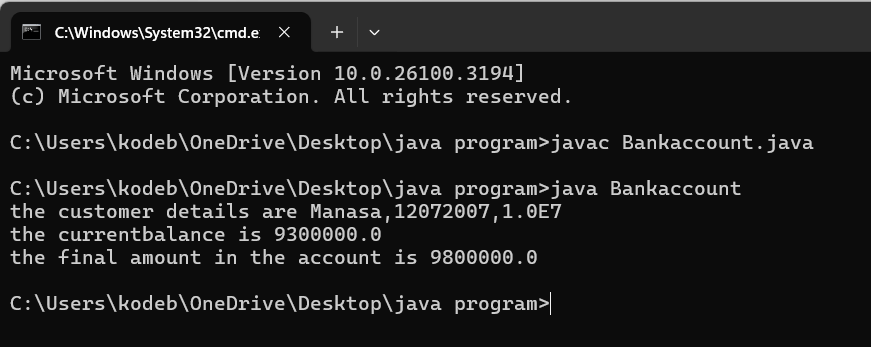
**CLASS DIAGRAM:**

Class Bankaccount

- AccName : String  
- AccNo : int  
- currentbalance : float

+ withdraw() : void  
+ deposit() : void

**OUTPUT:**

****

**ERRORS:**

|  |  |
| --- | --- |
| Errors Found | Errors Rectification |
| I typed the code as This.AccName = AccName ; | I rectified it as this.AccName = AccName; |
| I wrote the print statement as  System.out.println("the customer details are "+AccName+","+AccNo+","+currentbalance"); | I rectified the print statement as  System.out.println("the customer details are "+AccName+","+AccNo+","+currentbalance); |

**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 aconstructor with parameters which initializes title,author,yr of publication.Create a method which displays the details od three books.

**CODE:**

class Book{

private String author;

private String title;

private int year;

public Book(String author,String title,int year){

this.author=author;

this.title=title;

this.year=year;

}

public void author() {

System.out.println("The author of the book is"+author);

}

public void title() {

System.out.println("the title of the book is"+title);

}

public void year() {

System.out.println("the year of publication of a book is"+year);

}

public static void main(String[] args){

Book Book1 =new Book("George Orwell","Animal Farm",1945);

Book Book2 =new Book("Jane Austen","Pride And Prejudice",1813);

Book Book3 =new Book("Lewis Carroll","Alice’s Adventures In Wonderland",1865);

Book1.author();

Book1.title();

Book1.year();

Book2.author();

Book2.title();

Book2.year();

Book3.author();

Book3.title();

Book3.year();

}

}

**CLASS DIAGRAM:**

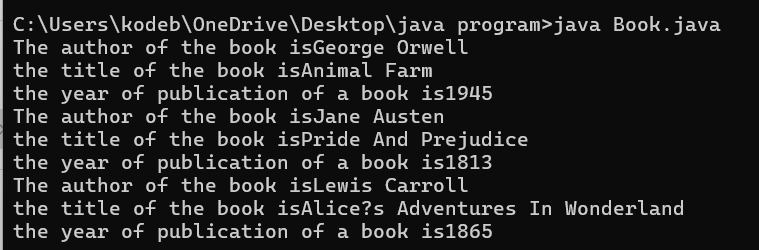
Class Book

- Title: String  
- Author : String  
- Year of Publication : int

+ Title() : void  
+ Author() : void

+Year of publication() :void

**OUTPUT:**

****

**ERRORS:**

|  |  |
| --- | --- |
| Errors Found | Errors Rectification |
| error: ';' expected  Book2.title() | Rectified by keeping ; at the end of Book2.title() |

**PROGRAM:2**

**AIM:**Create a Java program program with class name Myclass with a static variable count of int type initialised to zero and the constant variable “pi” of type double initialsed to 3.14 as attributes of the class .Now define a constructor for “Myclass” that increments the count variable each time an object my class in created .Finally print the final values of count &pi varables.Create 3 objects.

**CODE:**

class Myclass {

static int Count=0;

final static double pi = 3.14;

public Myclass(){

Count++;

}

public int Count(int objCount){

System.out.println("the count value of obj"+Count+" is "+Count);

System.out.println("the pi value is "+pi);

return Count;

}

public static void main(String[] args){

Myclass obj1 = new Myclass();

obj1.Count(0);

Myclass obj2 = new Myclass();

obj2.Count(1);

Myclass obj3 = new Myclass();

obj3.Count(2);

}

}

**CLASS DIAGRAM:**

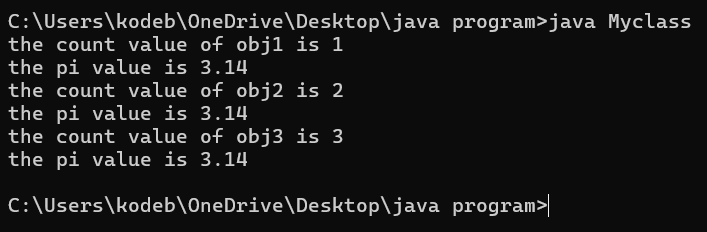
Class Myclass

- count : int

-pi : double

+Count() :int

**OUTPUT:**

****

**ERRORS:**

|  |  |
| --- | --- |
| Errors Found | Errors Rectification |
| error: ';' expected  Myclass obj3=new Myclass() | Rectified by keeping ; at the end of Myclass obj3=new Myclass() |

**IMPORTANT POINTS:**

1. The static variable is a variable shared among all instance of the class.
2. Syntax of static variable - **static datatype varname = value;**
3. Final keyword is used to fix the value to the variable.
4. Syntax to final a variable – **final datatype varname = value;**

**WEEK-5**

**PROGRAM-1:**

**AIM :** Create a calculator using the operations including addition, subtraction, multiplication, inheritance, and display the desired output.

**Code:**

class calculator {

protected double a, b;

public calculator(double a, double b) {

this.a = a;

this.b = b;

}

}

class Addition extends calculator {

public Addition(double a, double b) {

super(a, b);

}

public double add() {

return a + b;

}

}

class Subtraction extends Addition {

public Subtraction(double a, double b) {

super(a, b);

}

public double subtract() {

return a - b;

}

}

class Multiplication extends Subtraction {

public Multiplication(double a, double b) {

super(a, b);

}

public double multiply() {

return a \* b;

}

}

class Division extends Multiplication {

public Division(double a, double b) {

super(a, b);

}

public double divide() {

if (b != 0) {

return a / b;

} else {

System.out.println("Error");

return Double.NaN;

}

}

}

class Final extends Division {

public Final(double a, double b) {

super(a, b);

}

public void displayResults() {

System.out.println("Addition: " + add());

System.out.println("Subtraction: " + subtract());

System.out.println("Multiplication: " + multiply());

System.out.println("Division: " + divide());

}

}

import java.util.Scanner;

public class allcalculator {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.println("Enter a number: ");

        double a = input.nextDouble();

        System.out.println("Enter b number: ");

        double b = input.nextDouble();

        Final calc = new Final( a,  b);

        calc.displayResults();

        System.out.println("Manasa Kodeboyena");

        input.close();

    }}

**CLASS DIAGRAM:**

Calculator

-a: double

-b: double

+Calculator(a,b)

Addition

+add(): double

Subtraction

+subtract(): double

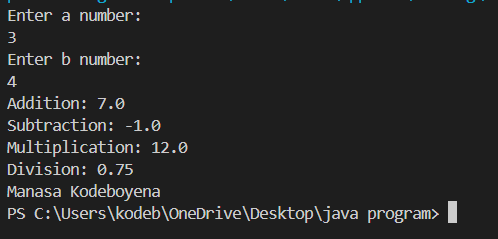
Multiplication

+multiply(): double

Division

+divide(): double

**Output:**

****

**ERROR:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. not providing the return method correctly. 2. Not mentioning super to obtain the super class constructor. | 1. After declaring methods, we must provide the return method correctly. 2. To obtain the super class we need to mention super. |

**Porgram:2**

**AIM**: : A vehicle rental company wants to develop a system that maintains information about different types of vechicles 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( should be in super class)

1. cars should have an additional property: no.of doors
2. Bikes should have a property indicating whether they have gears or not.
3. The system should also include a function to display details about each vehicle and indicate when a vehicle is starting.
4. Every class should have a constructor

Question:

1. Which oops concept is used in the above program
2. If the company decides to add a new type of vehicle, Truck, how would you modify the program?
3. Truck should include an additional property capacity (in tons)
4. Create a showTruckdetails() method to display the truck’s capacity.
5. Write a constructor for Truck that initializes all properties
6. Implement the truck class and update the main method to create a Truck object and also create an object for car and bike sub classes Finally, display the details.

**CODE:**

public class vehicle {

public String brand;

public int speed;

public vehicle(String brand, int speed) {

this.brand = brand;

this.speed = speed;

}

public void start() {

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

}

public void showDetails() {

System.out.println("Brand: " + brand);

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

}

}

class Car extends vehicle {

private int noOfDoors;

public Car(String brand, int speed, int noOfDoors) {

super(brand, speed);

this.noOfDoors = noOfDoors;

}

public void showDetails() {

super.showDetails();

System.out.println("Number of Doors: " + noOfDoors);

}

}

class Bike extends vehicle {

private boolean hasGears;

public Bike(String brand, int speed, boolean hasGears) {

super(brand, speed);

this.hasGears = hasGears;

}

public void showDetails() {

super.showDetails();

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

}

}

class Truck extends vehicle {

private int capacity;

public Truck(String brand, int speed, int capacity) {

super(brand, speed);

this.capacity = capacity;

}

public void showTruck() {

super.showDetails();

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

}

}

public class rent {

    public static void main(String[] args) {

        Car car = new Car("Toyota", 150, 4);

        Bike bike = new Bike("Yamaha", 120, true);

        Truck truck = new Truck("Volvo", 90, 10);

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

        car.start();

        car.showDetails();

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

        bike.start();

        bike.showDetails();

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

        truck.start();

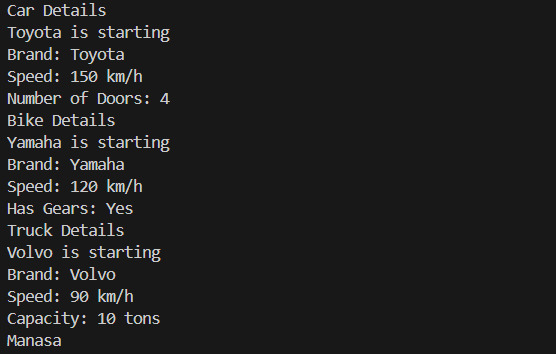
        truck.showTruck();

        System.out.println("Manasa”);

    }

}

**Output:**

****

**CLASS DIAGRAM:**

Vehcile

* Brand : str
* Speed : int

+ init(brand,speed)

+ start\_vehicle()

+display\_details()

Car

-no.of doors: int

+init (brand,speed displaydetails(), no.of doors)

+displaydetails()

+showtruckdetails()

Truck

-capacity: float

Bike

-has gears: bool

+init (brand, speed, has gears)

+displaydetails()

**ERRORS:**

|  |  |
| --- | --- |
| Code Error | Code rectification |
| 1. Declaring two superclasses inside the same file. 2. Not declaring the variable using ‘this’ keyword inside the constructor. | 1. Make two separate files to save the two super classes. 2. Declare the variable using this keyword to run the program. |