**23CSE111**

**OBJECT ORIENTED PROGRAMMING**

**LAB REPORT**



**Department of Computer Science Engineering**

**Amrita School of Computing**

**Amrita Vishwa Vidyapeetham, Amaravati Campus**

**Name: N.Abhishek**

**Verified By Roll No: 24232**

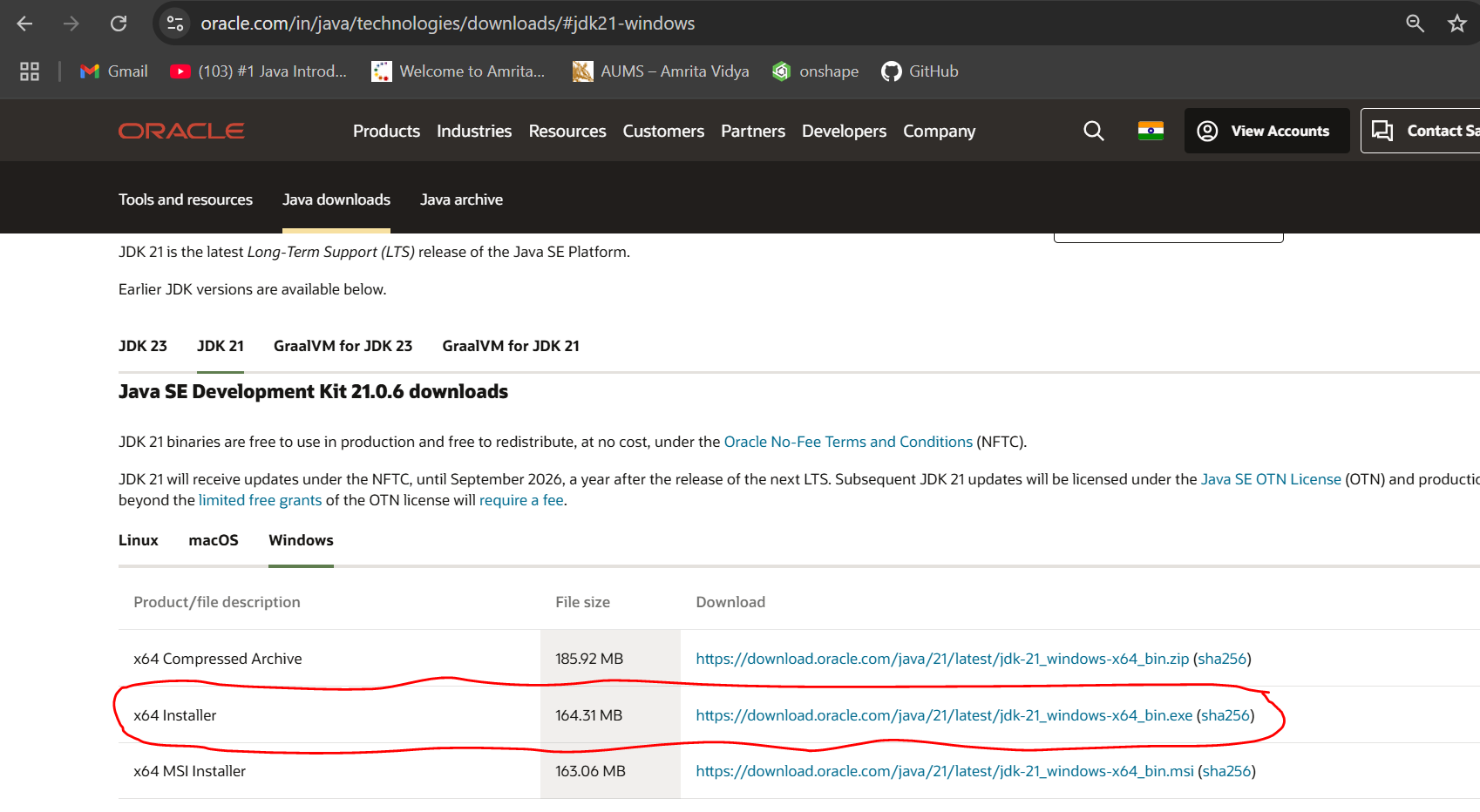
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Experiment Name** | **Page No.** | **Remarks** | **Signature** |
| 1 | Installation Process of JDK | 3 |  |  |
| 2 | Simple Java Program for printing basic details of student | 5 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**WEEK-1**

**1) Explain the process of Installing JDK (Java Development Kit)**

**Installing of JDK (Java Development Kit):**

1. **Download JDK:**
   * Go to the Oracle JDK download page in your web browser and click on JDK-21 version which is Long term support (LTS) version.
   * Click on the download link for your operating system (Windows, macOS, or Linux).



1. **Install JDK:**
   * Once downloaded, run the installer.
   * Follow the instructions and keep clicking "Next" until it's done.
2. **Set Environment Variables (Windows):**
   * For copying of path, go to c drive in file explorer
   * Then click on program files, in program files we can find the Java file click on it the in java file open bin file. Then click on the top of the file location then there we can select the file path location then copy it.
   * Open file explorer, then right click on This PC next select on properties then it will take you to the settings app then click on advanced system settings and then  
     click on **Environment Variables**.
   * Click **New** under **System Variables**:
     + **Set Variable name as:** java\_home
     + **Variable value:** The folder address where JDK is installed (like C:\Program Files\Java\jdk-21\bin)
   * Find Path under **System Variables**, click **Edit**, and add the path of the jdk-21(C:\Program Files\Java\jdk-21\bin)  
       
       
       
     

**Checking of JDK Version:**

1. **Open Command Prompt:**
   * Press win+R, type cmd, and press Enter.
2. **Check Version:**
   * Type java --version and press Enter.
   * Type javac --version and press Enter.



**2) Implement a Java Program to print basic details of a Student**

Write your code in Notepad and execute in cmd prompt

**Important Points:**

* + - * **Understand the syntax of a class.**

class Main {

public static void main(String[] args) {

System.out.println(" ");

}

}

**CODE:**

**The file was saved with Main.java**

class Main

{

public static void main(String[] args)

{

// Printing my basic details

System.out.println("Name:N.ABHISHEK");

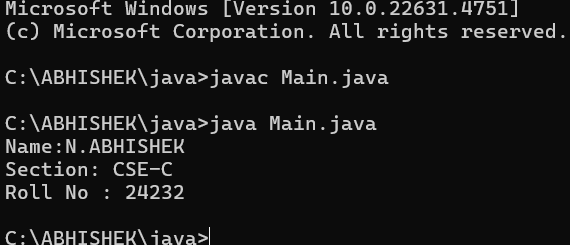
System.out.println("Section: CSE-C");

System.out.println("Roll No : 24232");

}

}

**Output:**

****

**Week 2**

1. **Simple Java Program for finding simple interest by taking input from**

**User**

**Code:**

**import java.util.Scanner;**

**public class Si {**

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

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

**// Taking user inputs**

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

**double principal = scanner.nextDouble();**

**System.out.print("Enter Rate of Interest (per annum): ");**

**double rate = scanner.nextDouble();**

**System.out.print("Enter Time (in years): ");**

**double time = scanner.nextDouble();**

**// Calculate Simple Interest**

**double simpleInterest = (principal \* rate \* time) / 100;**

**// Display Output**

**System.out.println("Principal Amount: " + principal);**

**System.out.println("Rate of Interest: " + rate + "% per annum");**

**System.out.println("Time Period: " + time + " years");**

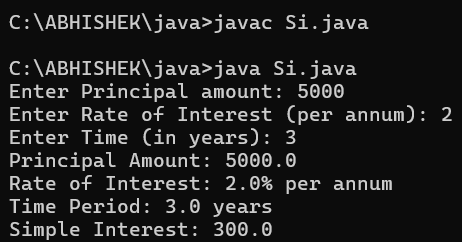
**System.out.println("Simple Interest: " + simpleInterest);**

**scanner.close();**

**}**

**}**

**Output**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **rectification** |
| **1** | **Syntax error** | **Did’nt declare variable type** | **Declared correct type** |
| **2** | **Syntax error** | **{ missing** | **{ added** |

**2.Write a simple program to calculate factorial of a number and read the**

**input from user**

**code:**

**import java.util.Scanner;**

**public class FactorialCalculator {**

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

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

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

**int number = scanner.nextInt();**

**long factorial = 1;**

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

**factorial \*= i;**

**}**

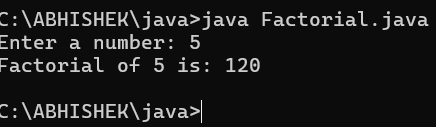
**System.out.println("Factorial of " + number + " is: " + factorial);**

**scanner.close();**

**}**

**}**

**Output:**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| **1** | **Missing import statement** | **Not importing packages** | **Packages imported** |
| **2** | **Logical error** | **Wrong formula** | **Formula rectified** |

**3.Write a program to to calculate the fibonacii sequence and take the input from user**

**Code:**

**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);**

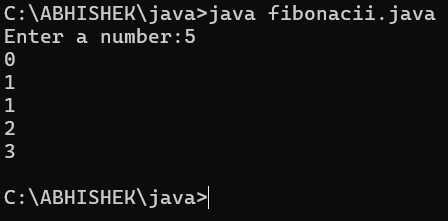
**i = i+1;**

**}**

**}**

**}**

**Output:**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| **1** | **Logical error** | **Incorrect formula** | **Formula rectified** |
| **2** | **Run-time error** | **Incorrect path** | **Added correct path** |

**Write a java program to convert temperature from Fahrenheit to celsius**

**Code:**

**import java.util.Scanner;**

**public class TemperatureConverter {**

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

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

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

**double fahrenheit = scanner.nextDouble();**

**double celsius = (fahrenheit - 32) \* 5 / 9;**

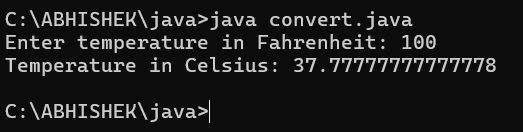
**System.out.println("Temperature in Celsius: " + celsius);**

**scanner.close();**

**}**

**}**

**Output:**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **rectification** |
| **1** | **Syntax error** | **Missing ”** | **“ is added** |
| **2** | **Missing variable type** | **Didn’t declare variable** | **Declared variable** |

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

**Code:**

**import java.util.Scanner;**

**public class convert {**

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

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

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

**double fahrenheit = scanner.nextDouble();**

**double celsius = (fahrenheit - 32) \* 5 / 9;**

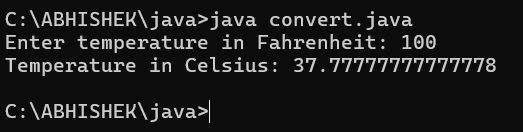
**System.out.println("Temperature in Celsius: " + celsius);**

**scanner.close();**

**}**

**}**

**Output:**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| **1** | **Runtime error** | **Incorrect path selection** | **Correct path added** |
| **2** | **Logical error** | **Incorrect logic** | **Correct logic** |

**Write a simple program to find the area of rectangle:**

**Code:**

**import java.util.Scanner;**

**public class Area {**

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

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

**System.out.print("Enter the length of the rectangle: ");**

**double length = scanner.nextDouble();**

**System.out.print("Enter the width of the rectangle: ");**

**double width = scanner.nextDouble();**

**double area = length \* width;**

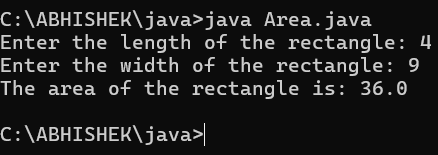
**System.out.println("The area of the rectangle is: " + area);**

**scanner.close();**

**}**

**}**

**Output:**



|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| **1** | **Syntax error** | **Semi colon missing** | **Semi colon added** |
| **2** | **Missing import error** | **Import package missing** | **Import package added** |
| **3** |  |  |  |

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

**Code:**

import java.util.Scanner;

public class TriangleAreaCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the first side of the triangle: ");

double a = scanner.nextDouble();

System.out.print("Enter the second side of the triangle: ");

double b = scanner.nextDouble();

System.out.print("Enter the third side of the triangle: ");

double c = scanner.nextDouble();

double s = (a + b + c) / 2;

double area = Math.sqrt(s \* (s - a) \* (s - b) \* (s - c));

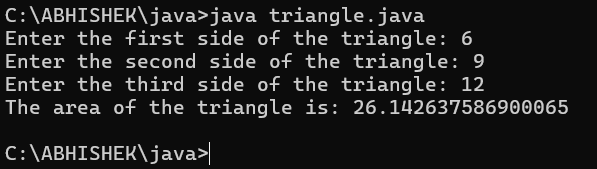
System.out.println("The area of the triangle is: " + area);

scanner.close();

}

}

**OUTPUT:**



|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| **1** | **Logical error** | **Incorrect formula** | **Formula rectified** |
| **2** | **Name error** | **Undeclared variable** | **Variable declared** |
|  |  |  |  |

WEEK – 3

1.Write a code for method attributes of a car for three objects.

Aim:writing java code using method attributes.

Code:

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="BMW";

car1.fuel\_type="Petrol";

car1.mileage=62.5F;

//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="Audi";

car2.fuel\_type="Petrol";

car2.mileage=64.5F;

// 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="Yellow";

car3.car\_brand="Benz";

car3.fuel\_type="Diesel";

car3.mileage=66.5F;

// 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);

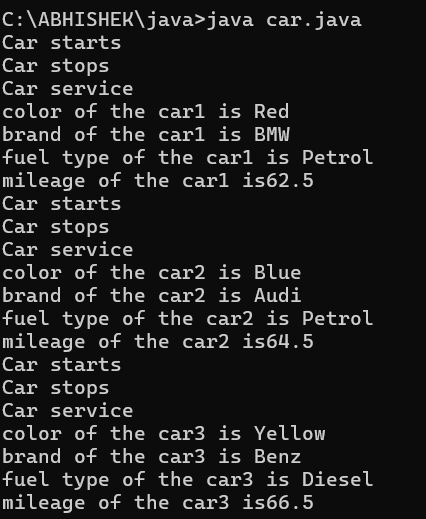
System.out.println("fuel type of the car3 is "+car3.fuel\_type);

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

}

}

OUTPUT:



|  |  |  |  |
| --- | --- | --- | --- |
| Error type | Description | impact | Solution |
| Syntax error | Typo in the variable name mileage in the constructor | Compilation error ;  Constructor won’t work | Corrected to mileage |
| Mthod signature | Getter methods color,brand,fuel,mile had parameters but were expected to return instance values | Causes incorrect behaviour when  Calling methods | Removed parameters from method definition |

CLASS DIAGRAM :

|  |
| --- |
| Main |
| -car\_color:string  -car\_brand: string  - fuel\_type:string  -mileage:int |
| +color():String  +brand():String  +fuel():String  +mile():int |

Aim:writing a java code for the bank account details

Code:

class Bank\_Account {

private String accname;

private int acno;

private float balance;

public Bank\_Account(String accname, int acno, float balance) {

this.accname = accname;

this.acno = acno;

this.balance = balance;

}

public void withdraw(int amount) {

if (amount <= balance) {

balance -= amount;

System.out.println("Withdrawal of " + amount + " successful. Remaining balance: " + balance);

} else {

System.out.println("Insufficient balance for " + accname);

}

}

public void deposit(int amount) {

balance += amount;

System.out.println("Deposit of " + amount + " successful. Updated balance: " + balance);

}

public void displayDetails() {

System.out.println("Account Name: " + accname);

System.out.println("Account Number: " + acno);

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

}

public static void main(String[] args) {

Bank\_Account s = new Bank\_Account("Anil", 34571, 25000);

s.withdraw(20000);

s.deposit(2000);

s.displayDetails();

System.out.println();

Bank\_Account s1 = new Bank\_Account("Jeevan", 333226, 25000);

s1.withdraw(2500);

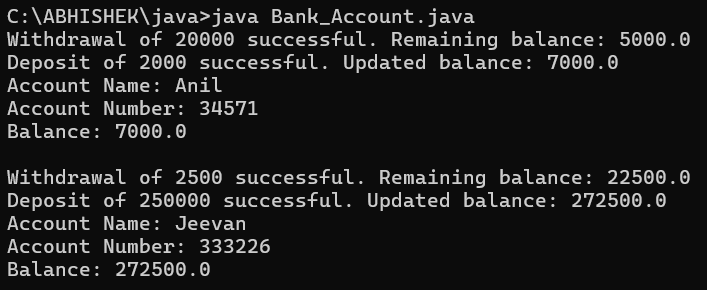
s1.deposit(250000);

s1.displayDetails();

}

}

Output:



|  |  |  |  |
| --- | --- | --- | --- |
| Error type | subscription | Impact | Solution |
| Missing balance dispaly | Withdraw and deposit methods update the balance but not dispaly | User cannot see the current balance after transactions | Add system.out.println on both methods |
| No getter methods | No method to retrieve account details | Cannot account details outside of the class | Add getter methods like getBalance(), getAccName(), etc. |

CLASS DIAGRAM:

|  |
| --- |
| Bank\_Account |
| -accname:String  -acno:int  -balance:float |
| +withdraw(amount:int):void  +deposit(amount:int):void  +displaydetails():void |

WEEK 4:

LAB 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.

CODE:

public class Book {

public String title;

public String author;

public int year;

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

this.title = title;

this.author = author;

this.year = year;

}

public void displayDetails() {

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

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

System.out.println("Year of Publication" +year);

}

public static void main(String[] args) {

Book b1 = new Book("Math", "Ramanujan", 1950);

Book b2 = new Book("Physics", "CV Raman", 1960);

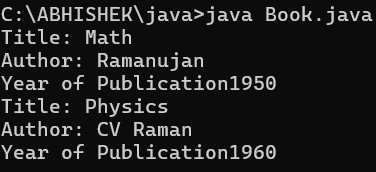
b1.displayDetails();

b2.displayDetails();

}

}

OUTPUT:



|  |  |  |  |
| --- | --- | --- | --- |
| ERROR TYPE | DESCRIPTION | IMPACT | SOLUTION |
| Compilation error | Missing semicolon | Code will not compile | Add semicolons at the end of the lines |
| Logical error | Missing spacing in the print statements | Output may be concatenated improperly | Add a space after year of publication in the display details () method |

CLASS DIAGRAM:

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

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.

CODE:

public class MyClass {

static int count = 0;

static final double pi = 3.14;

MyClass() {

count++;

}

public static void main(String[] args) {

MyClass obj1 = new MyClass();

MyClass obj2 = new MyClass();

MyClass obj3 = new MyClass();

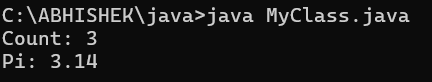
System.out.println("Count: " +count);

System.out.println("Pi: " +pi);

}

}

OUTPUT:



|  |  |  |  |
| --- | --- | --- | --- |
| Error type | Description | Impact | Solution |
| Logical error | Missing space in the println statement for count and pi | Output may be concatenated improperly. | Add a space before +count and+pi in the println statement. |

CLASS DIAGRAM:

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

**WEEK-5**

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

**Program :**

import java.util.Scanner;

class Addition {

int add(int a, int b) {

return a + b;

}

}

class Subtraction extends Addition {

int subtract(int a, int b) {

return a - b;

}

}

class Multiplication extends Subtraction {

int multiply(int a, int b) {

return a \* b;

}

}

class Division extends Multiplication {

double divide(int a, int b) {

if (b != 0) {

return (double) a / b;

} else {

return Double.NaN;

}

}

}

public class Calculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Division calc = new Division();

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

int num1 = scanner.nextInt();

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

int num2 = scanner.nextInt();

System.out.println("Addition: " + calc.add(num1, num2));

System.out.println("Subtraction: " + calc.subtract(num1, num2));

System.out.println("Multiplication: " + calc.multiply(num1, num2));

double divisionResult = calc.divide(num1, num2);

if (Double.isNaN(divisionResult)) {

System.out.println("Division: Error! Division by zero.");

} else {

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

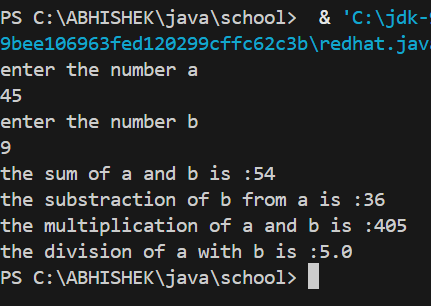
}

scanner.close();

}

}

**OUTPUT:**

****

**ERROR :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| 1 | syntax error | String forgot in main function | String is added |
| 2 | Logical error | Incorrect logic | Correct logic |

**2)** **Aim:** 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 variable such as brand and speed.

• Cars should have an additional property: numbers 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 & indicates when a vehicle is starting.

• Every class should have constructor.

1. Which OOP concept is used in the above program? Explain why it is useful in this scenario.

2. If the company decides to add a new type of vehicle: Truck, how would you modify the program?

• Truck should include an additional property capacity(in tons)

• Create a ShowTruckDetails() method to display the truck’s capacity.

• Write a constructor for Truck that initializes all properties.

3)Aim: Implement the truck class and update the main method to create a truck object & also create an object for car & bike sub classed. Finally display its details

**Program :**

import java.util.Scanner;

class Vehicle {

String brand;

int speed;

Vehicle(String brand, int speed) {

this.brand = brand;

this.speed = speed;

}

void start() {

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

}

void showDetails() {

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

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

}

}

class Car extends Vehicle {

int numberOfDoors;

int seatingCapacity;

Car(String brand, int speed, int numberOfDoors, int seatingCapacity) {

super(brand, speed);

this.numberOfDoors = numberOfDoors;

this.seatingCapacity = seatingCapacity;

}

void showCarDetails() {

showDetails();

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

System.out.println("Seating Capacity: " + seatingCapacity);

}

}

class Bike extends Vehicle {

boolean hasGears;

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

super(brand, speed);

this.hasGears = hasGears;

}

void showBikeDetails() {

showDetails();

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

}

}

class Truck extends Vehicle {

double capacity;

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

super(brand, speed);

this.capacity = capacity;

}

void showTruckDetails() {

showDetails();

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

}

}

public class RentalSystem {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Car car = new Car("Toyota", 180, 4, 5);

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

Truck truck = new Truck("Volvo", 100, 15.5);

car.start();

car.showCarDetails();

System.out.println();

bike.start();

bike.showBikeDetails();

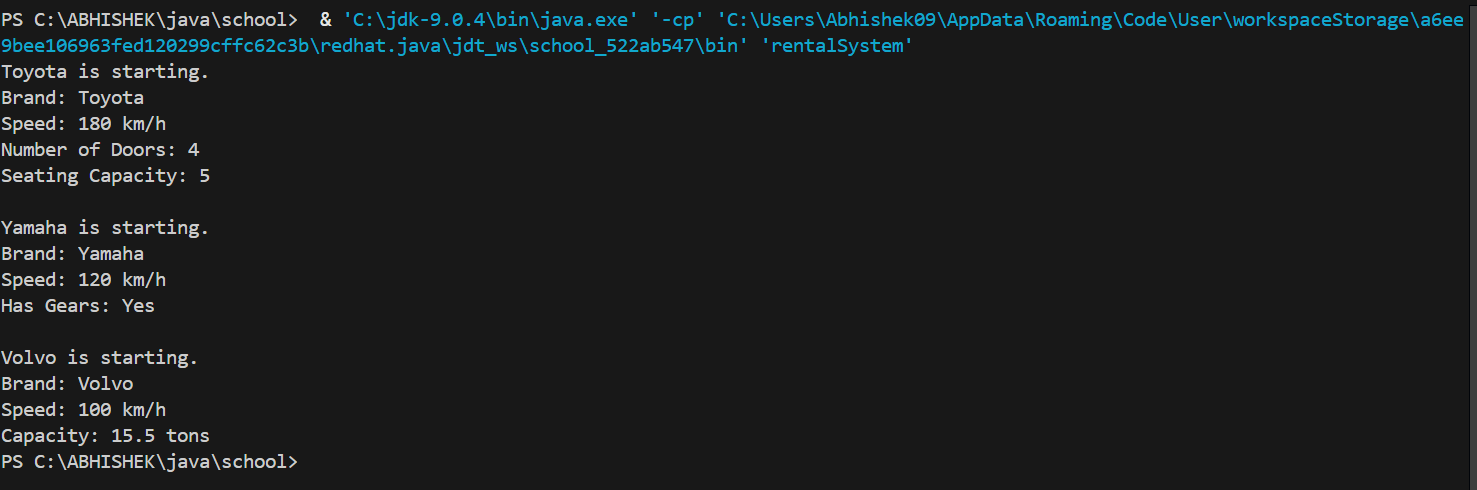
System.out.println();

truck.start();

truck.showTruckDetails();

scanner.close();

}

}**OUTPUT :** ****

**ERROR :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| 1 | syntax error | String forgot in main function | String is added |
| 2 | Logical error | Incorrect logic | Correct logic |

**WEEK - 6**

1. **Aim : Write a Java program to create a Vehicle class with a method displayInfo(). Override this method in the Car subclass to provide specific information about a car.**

**Program :**

class Vehicle {

void displayInfo() {

System.out.println("This is a vehicle.");

}

}

class Car extends Vehicle {

@Override

void displayInfo() {

System.out.println("This is a car. Cars are comfortable and fast.");

}

}

public class carInfo {

public static void main(String[] args) {

Vehicle genericVehicle = new Vehicle();

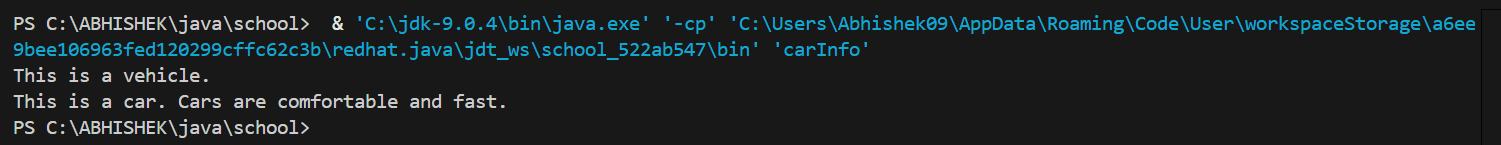
genericVehicle.displayInfo();

Car car = new Car();

car.displayInfo();

}

}**OUTPUT :**

****

**ERRORS :**

**No errors**

**2) Aim :** A college is developing an automated admission system that verifies students' eligibility for undergraduate (UG) and postgraduate (PG) programs. Each program has different eligibility criteria based on the students' percentage in their previous qualifications.

(i)UG admissions require a minimum of 60%.

(ii)PG admissions require a minimum of 70%

**Program :**

import java.util.Scanner;

class Admission {

void checkEligibility(double percentage) {

System.out.println("Checking general eligibility.");

}

}

class UGAdmission extends Admission {

@Override

void checkEligibility(double percentage) {

if (percentage >= 60) {

System.out.println("Eligible for UG program.");

} else {

System.out.println("Not eligible for UG program.");

}

}

}

class PGAdmission extends Admission {

@Override

void checkEligibility(double percentage) {

if (percentage >= 70) {

System.out.println("Eligible for PG program.");

} else {

System.out.println("Not eligible for PG program.");

}

}

}

public class admissionSystem {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter your percentage for UG admission: ");

double ugPercentage = scanner.nextDouble();

UGAdmission ug = new UGAdmission();

ug.checkEligibility(ugPercentage);

System.out.print("Enter your percentage for PG admission: ");

double pgPercentage = scanner.nextDouble();

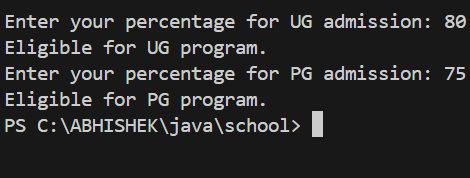
PGAdmission pg = new PGAdmission();

pg.checkEligibility(pgPercentage);

scanner.close();

}

}**OUPUT :**

****

**ERROR :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| 1 | syntax error | String forgot in main function | String is added |
| 2 | Logical error | Incorrect logic | Correct logic |

**3) Aim :** Create a Calculator class with overloaded methods to perform addition:

(i) Add two integers.

(ii) Add two doubles.

(iii) Add three integers.

**Program :**

class Calculator1 {

int add(int a, int b) {

return a + b;

}

double add(double a, double b) {

return a + b;

}

int add(int a, int b, int c) {

return a + b + c;

}

public static void main(String[] args) {

Calculator1 calc = new Calculator1();

int sumTwoInts = calc.add(5, 10);

double sumTwoDoubles = calc.add(5.5, 10.5);

int sumThreeInts = calc.add(1, 2, 3);

System.out.println("Sum of two integers: " + sumTwoInts);

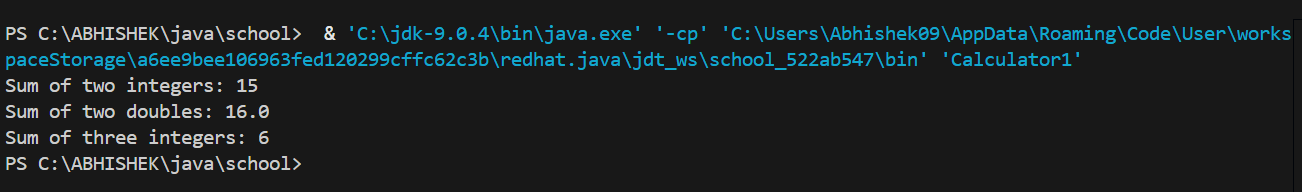
System.out.println("Sum of two doubles: " + sumTwoDoubles);

System.out.println("Sum of three integers: " + sumThreeInts);

}

}

**OUTPUT :**

****

**ERROR :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| 1 | syntax error | String forgot in main function | String is added |
| 2 | Logical error | Incorrect logic | Correct logic |

**4)Aim :** Create a Shape class with a method calculateArea() that is overloaded for different shapes (e.g., square, rectangle). Then, create a subclass Circle that overrides the calculateArea() method for a circle.

**Program :**

class Shape {

double calculateArea(double side) {

return side \* side;

}

double calculateArea(double length, double breadth) {

return length \* breadth;

}

}

class Circle extends Shape {

@Override

double calculateArea(double radius) {

return Math.PI \* radius \* radius;

}

public static void main(String[] args) {

Shape shape = new Shape();

Circle circle = new Circle();

double squareArea = shape.calculateArea(5);

double rectangleArea = shape.calculateArea(4, 6);

double circleArea = circle.calculateArea(3);

System.out.println("Area of Square: " + squareArea);

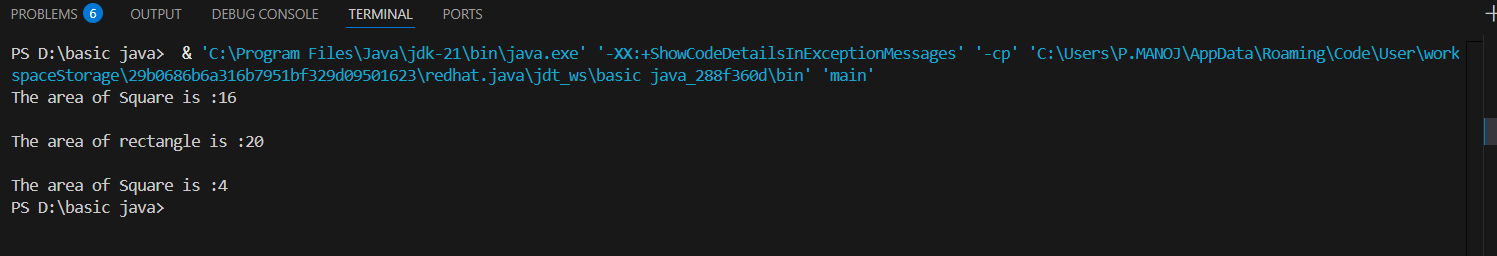
System.out.println("Area of Rectangle: " + rectangleArea);

System.out.println("Area of Circle: " + circleArea);

}

}

**OUTPUT :**

****

**ERRORS :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| 1 | Syntax error | ; is missed | ; is added |
| 2 | Logical error | Incorrect logic | Correct logic |