­­  **23CSE111**

**OBJECT ORIENTED PROGRAMMING**

**LAB REPORT**



**Department computer science and engineering**

**Amrita school of computing**

**Amrita vishwa vidyapeetham, Amaravati campus**

Name: K. Divya teja

Verified by Class/section : CSE – B

Roll no:av.sc.u4cse24138

INDEX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S no | **Title** | **Page no** | **Date** | **Signature** |
|  | **WEEK -1** |  |  |  |
| **1** | Explain the process of download & Installation of JDK | **4** |  |  |
| **2** | Write a program to print Student’s name, Roll no, Section | **7** |  |  |
|  | **WEEK-2** |  |  |  |
| **1** | Write a simple java program to calculate factorial of a number | **9** |  |  |
| **2** | Write a simple java program to find the simple interest by taking iinputs from the user | **10** |  |  |
| **3** | Write a program to calculate the Fibonacci sequence and take the input from the user | **13** |  |  |
| **4** | Write a program to find the area of triangle using hereon’s formula  . | **15** |  |  |
| **5** | Write a program to convert temperature from celssius to fahrenheit | **17** |  |  |
|  | **WEEK-3** |  |  |  |
| **1** | Write a java program with following instructions | **19** |  |  |
| **2** | Write a java program with following instructions | **26** |  |  |
|  | **WEEK-4** |  |  |  |
| **1** | Write a java program with following instructions | **34** |  |  |
| **2** | Write a java program with following instructions | **38** |  |  |
|  | WEEK-5 |  |  |  |
| **1** | Write a java program with following instructions |  |  |  |
| **2** | Write a java program with following instructions |  |  |  |
|  | WEEK -6 |  |  |  |
| **1** | Write a java program with following instructions |  |  |  |
| **2** | Write a java program with following instructions |  |  |  |
| **3** | Write a java program with following instructions |  |  |  |
| **4** | Write a java program with following instructions |  |  |  |

**WEEK-1**

**1) AIM:**

**The process of download & installation of JDK(java development kit)**

**Download the JDK:**

1. Visit the official Oracle website and click on JDK 21 version and select our operating system type of your Desktop .
2. Click the link to download file description /product.

**Install JDK:**

1. Once downloaded, run the installer.

2. Follow the instructions and keep clicking "Next" until it's done.

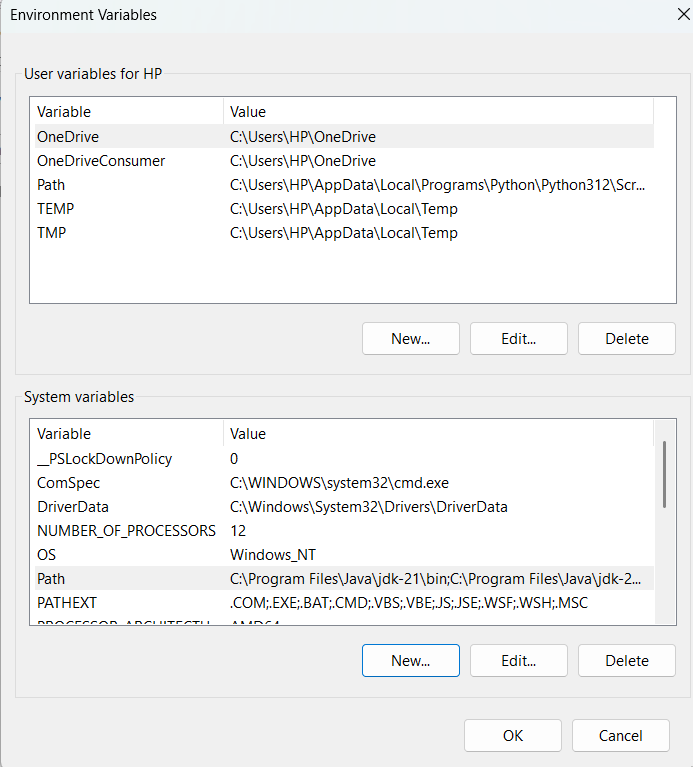
**Set Environment Variables (Windows):**

1. Open the Start Menu and search for "Environment Variables".

2. Click "Edit the system environment variables".

3. In the System Properties window, click "Environment Variables".

4. Under "System variables," Double click the path variable click the "New" and Add the bin directory of your JDK (C:\Program Files\Java\jdk-21\bin) to the path.

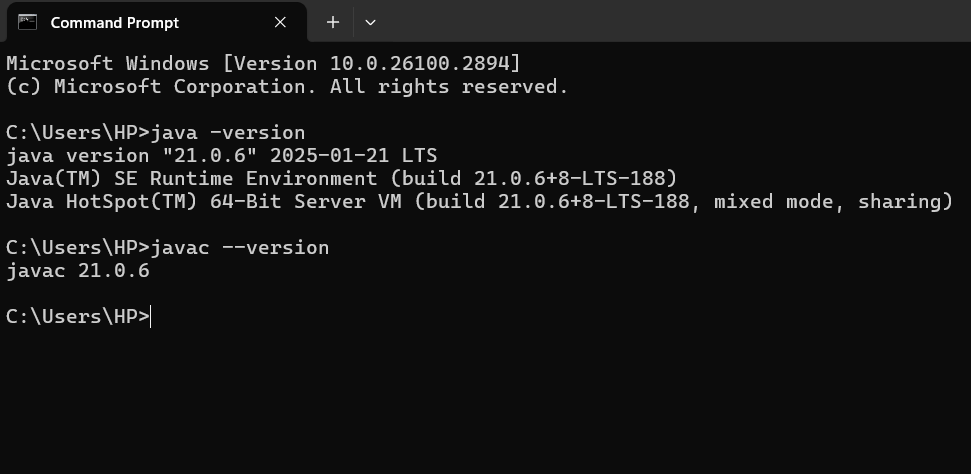
****

**Checking the JDK version:**

1. Open the start menu and search command prompt open it.
2. To check the java version

Type java –version and click enter (or)

Type javac –version and click enter to check the version



**2) AIM: Write a program to print Student’s name, Roll no and Section.**

Write our program in notepad and execute in Command prompt

**Program:**

class Demo

{

public static void main(String[] args)

{

System.out.println("NAME:k.Divya teja");

System.out.println("ROLL NO:av.sc.u4cse24148");

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

}

}

**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Error** | **Error rectification** |
| 1 | path error | Using correct path and  Absence of ‘-‘ symbol |
| 2 | Syntax error | Use Captial S in System.out.println() |

**WEEK-2**

**1) AIM:**

**Write a simple program to calculate factorial of a number and read the input from user**

**PROGRAM :**

class Test {

static int factorial(int n)

{

int res = 1, i;

for (i = 2; i <= n; i++)

res \*= i;

return res;

}

public static void main(String[] args)

{

int num = 5;

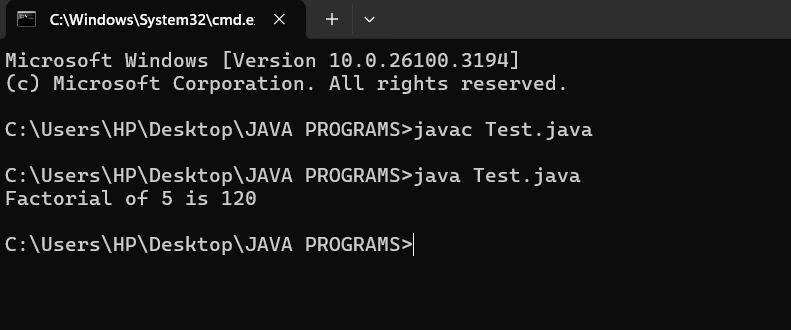
System.out.println("Factorial of " + num + " is "

+ factorial(5));

}

}

**OUTPUT :**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| 1 | Undeclared variable error | Missing variable | Variable declared |
| 2 | Missing import statement | Not importing packages | Packages imported |
| 3 | Logical error | Wrong formula | Formula rectified |

**2) AIM : Simple Java Program for finding simple interest by taking input from**

**PROGRAM :**

import java.util.Scanner;

class simple {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int p = input.nextInt();

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

int t = input.nextInt();

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

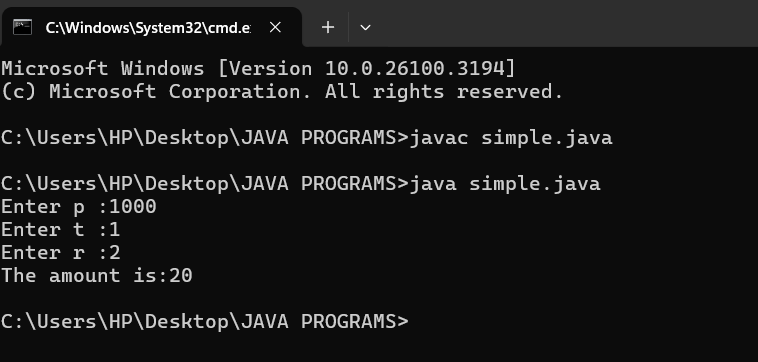
int r = input.nextInt();

System.out.println("The amount is:" + (p\*t\*r)/100);

}

}

**OUTPUT :**

****

**ERROR :**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

**3) AIM : Write a program to to calculate the fibonacii sequence and take the input from user**

**PROGRAM :**

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);

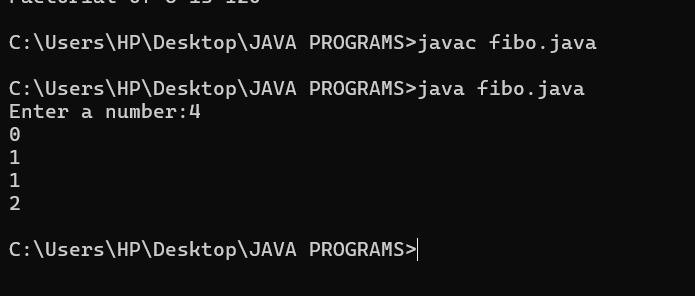
i = i+1;

}

}

}

**OUTPUT :**

****

**ERROR :**

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

**4) AIM : Write a java program to find area of triangle using heron’s formula and area of triangle**

**PROGRAM :**

import java.util.Scanner;

class Area {

public static void main(String args[]) {

Scanner input= new Scanner(System.in);

System.out.print("Enter the length of side a: ");

double a = input.nextDouble();

System.out.print("Enter the length of side b: ");

double b = input.nextDouble();

System.out.print("Enter the length of side c: ");

double c = input.nextDouble();

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

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

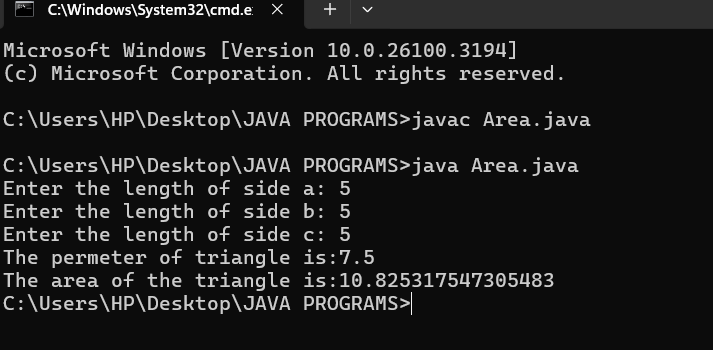
System.out.println("The permeter of triangle is:" + s);

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

}

}

**OUTPUT :**

****

**ERRORS :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S no** | **Error type** | **Reason for error** | **Rectification** |
| 1 | Syntax error | Semicolon missing | Semi colon added |
| 2 | Missing Scanner | Creating scanner input | Scanner added |

**5) AIM : Write a java program to convert temperature from Celsius to Fahrenheit**

**PROGRAM :**

class celsiustofahrenheit {

    public static void main(String[] args)

    {

        double celsius = 10.0, fahrenheit = 0.0;

        fahrenheit = (celsius \* 1.8) + 32;

        System.out.println(

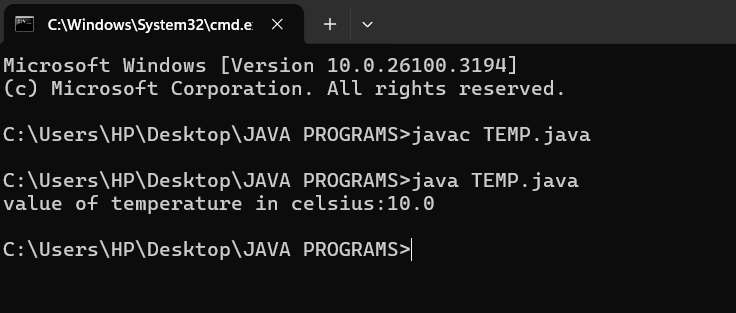
            " value of temperature in fahrenheit:"

            + fahrenheit);

    }

}

**OUTPUT :**

****

**ERRORS :**

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

**WEEK-3**

1. **AIM :**

**Write a java program with following instructions**

1. **Create a class with name car**
2. **Create four attributes named, car colour, car brand , fuel type , mileage.**
3. **Create 3 methods named start , stop, service .**
4. **Create 3 objects named c1,c2,c3**
5. **Create a constructor with parameters , car colour , car brand , fuel type and mileage.**

**PROGRAM :**

class car {

//ATTRIBUTES

String car\_brand;

String car\_colour;

String fuel\_type;

int mileage;

//CONSTRUCTORS

public car(String car\_brand,String car\_colour,String fuel\_type,int mileage){

this.car\_brand = car\_brand;

this.car\_colour = car\_colour;

this.fuel\_type = fuel\_type;

this.mileage = mileage;

}

//METHODS

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 I SERVICE");

}

public void car\_details() {

System.out.println("CAR BRAND IS :" + car\_brand);

System.out.println("CAR COLOUR IS :" + car\_colour);

System.out.println("CAR FUEL TYPE IS :" + fuel\_type);

System.out.println("CAR MILEAGE IS :" + mileage);

}

//MAIN PROGRAM

public static void main(String[] args) {

// CREATING OBJECTS FOR CLASS CAR

car c1 = new car("MCLAREN","GREEN","PETROL",12);

car c2 = new car("FERRARI","RED","PETROL",18);

car c3 = new car("LAMBORGHINI","ORANGE","PETROL",20);

//CALLING METHODS

c1.car\_details();

System.out.println(" ");

c2.car\_details();

System.out.println(" ");

c3.car\_details();

}

}class car {

//ATTRIBUTES

String car\_brand;

String car\_colour;

String fuel\_type;

int mileage;

//CONSTRUCTORS

public car(String car\_brand,String car\_colour,String fuel\_type,int mileage){

this.car\_brand = car\_brand;

this.car\_colour = car\_colour;

this.fuel\_type = fuel\_type;

this.mileage = mileage;

}

//METHODS

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 I SERVICE");

}

public void car\_details() {

System.out.println("CAR BRAND IS :" + car\_brand);

System.out.println("CAR COLOUR IS :" + car\_colour);

System.out.println("CAR FUEL TYPE IS :" + fuel\_type);

System.out.println("CAR MILEAGE IS :" + mileage);

}

//MAIN PROGRAM

public static void main(String[] args) {

// CREATING OBJECTS FOR CLASS CAR

car c1 = new car("MCLAREN","GREEN","PETROL",12);

car c2 = new car("FERRARI","RED","PETROL",18);

car c3 = new car("LAMBORGHINI","ORANGE","PETROL",20);

//CALLING METHODS

c1.car\_details();

System.out.println(" ");

c2.car\_details();

System.out.println(" ");

c3.car\_details();

}

}

**CLASS DIAGRAM :**

|  |
| --- |
| **CAR** |
| car\_brand : String  car\_colour : String  fuel\_type : String  mileage : int |
| + car (String, String, String,int)  +start() : void  +stop() : void  +service() : void  + car\_details : void |

**OUTPUT :**

****

**ERROR :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| 1 | Runtime error | Incorrect symbol in main program | { symbol is added |
| 2 | Logical error | Incorrect logic | Correct logic |

**4) AIM :**

**create a class named bankaccount with method deposit and with draw where the deposit method should accepts a parameter and when this method is called the deposited amount should be current balance .In addition to that when a withdraw method is called it has to verify whether withdraw amount is less than the current balance .If not display a message saying insufficient funds.Use the constructer to display the details of the customer (Customer name,account number , IFSC,branch) .Also create two customer objects c1,c2**

**PROGRAM :**

class bank {

    String name;

    String number;

    String IFSC;

    String branch;

    int balance;

//CONSTRUCTORS

public bank(String name, String number, String IFSC, String branch, int balance) {

    this.name = name;

    this.number = number;

    this.IFSC = IFSC;

    this.branch = branch;

    this.balance = balance;

}

//METHODS

public void bank\_details() {

        System.out.println("Customer Name: " + name);

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

        System.out.println("IFSC Code: " + IFSC);

        System.out.println("Branch: " + branch);

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

    }

//METHOD FOR DEPOSIT

public void deposit(int amount) {

    if (amount >0) {

    balance += amount;

    System.out.println("Total balance is :" + balance);

    } else {

    System.out.println("ERROR");

    }

}

//METHOD FOR WITHDRAWAL

public void withdrawal(int amount) {

    if (amount < balance) {

        balance -= amount;

        System.out.println("WITHDRAW AMOUNT IS:" + amount);

        System.out.println("UPDATED BALANCE IS: " + balance);

        } else {

            System.out.println("INSUFFICIENT BALANCE");

        }

    }

//MAIN FUNCTION

public static void main(String[] args) {

    //CREATING OBJECTS FOR CLASS BANK

    bank c1 = new bank("Rahul", "1234567890", "SBI1997","GUNTUR",0);

    bank c2 = new bank("Iyer", "1234567890", "SBI1996","GUNTUR",0);

    //CALLING METHODS

    //CUSTOMER 1 DETAILS

    System.out.println("CUSTOMER 1 DETAILS");

    System.out.println("    ");

    c1.bank\_details();

    System.out.println("    ");

    System.out.println("CUSTOMER 1 DEPOSIT");

    c1.deposit(1000);

    System.out.println("    ");

    System.out.println("CUSTOMER 1 WITHDRAW");

    c1.withdrawal(500);

    System.out.println("    ");

    System.out.println("    ");

    //CUSTOMER 2 DETAILS

    System.out.println("CUSTOMER 2 DETAILS");

    System.out.println("    ");

    c2.bank\_details();

    System.out.println("    ");

    System.out.println("CUSTOMER 2 DEPOSIT");

    c2.deposit(5000);

    System.out.println("    ");

    System.out.println("CUSTOMER 2 WITHDRAW");

    c2.withdrawal(600);

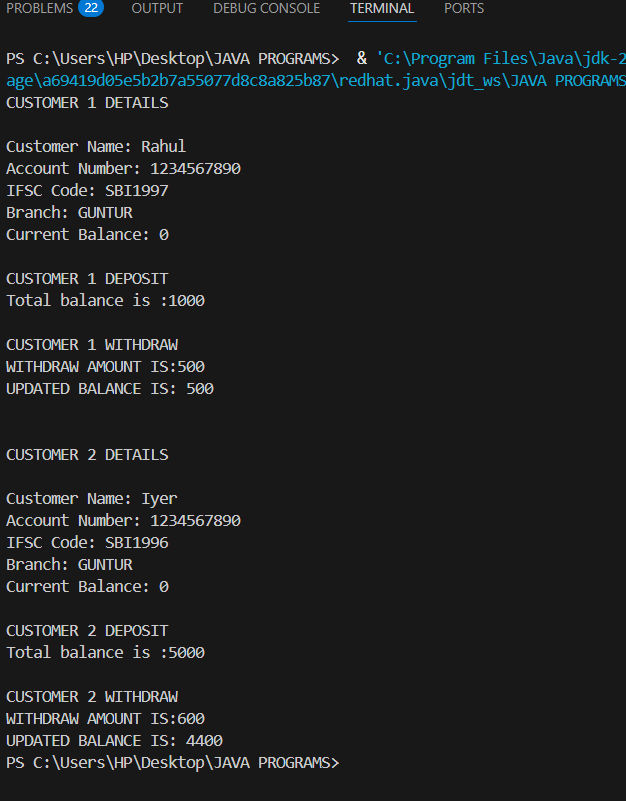
}

}

**CLASS DIAGRAM :**

|  |
| --- |
| bank |
| name:String  number:String  IFSC:String  branch:String  balance:int |
| bank(name:String , number:String, IFSC:String, +branch:String, balance:int)  +bank\_details():void  +deposit(amount:int):void  +withdrawl(amount:int):void |

**OUTPUT :**

****

**ERROR :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| 1 | syntax error | Forgot to keep main word in main program | Main is added |
| 2 | Logical error | Incorrect logic | Correct logic |

**WEEK-4**

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 with initializes title ,author and year of publication . create a method which displays the details of the book (display the details of two book i.e, create two books and objects with details).**

**PROGRAM:**

class book {

    String title;

    String author;

    int year;

//CONSTRUCTORS

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

    this.title = title;

    this.author = author;

    this.year = year;

}

//METHODS

public void details() {

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

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

    System.out.println("year of publication: " + year);

}

//MAIN FUNCTION

public static void main(String[] args) {

    //CREATING OBJECTS

    book b1 = new book("OPERATING SYSTEMS", "Galvin & Gange", 1997);

    book b2 = new book("DATABASE SYSTEMS", "Ramez Elmasri", 2004);

    //CALLING METHODS

    System.out.println("BOOK 1 DETAILS");

    System.out.println("    ");

    b1.details();

 System.out.println("    ");

    System.out.println("BOOK 2 DETAILS");

 System.out.println("    ");

    System.out.println("    ");

    b2.details();

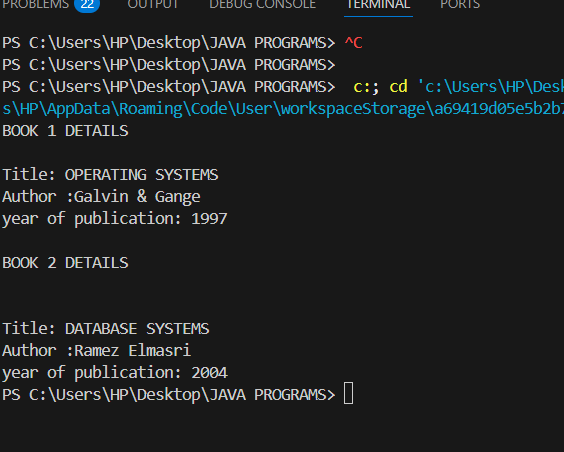
}

}

**CLASS DIAGRAM :**

|  |
| --- |
| book |
| title : String  author : String  year : int |
| + book(title: String, author: String, year: int)  + details(): void |

**OUTPUT :**

****

**ERROR :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| 1 | syntax error | Forgot to keep }  At last | } is added |
| 2 | Logical error | Incorrect logic | Correct logic |

1. **AIM : write a java program to create a class named myclass with a static variable count of int type and initialize to zero and a constant variable pie of double data type ,initialize to 3.1415 as attributes of that class now define a constructor for my class that increments the count variables each time an object of my class is created variable each time an object of myclass is created. Finally print the final values of count and pie variables.**

**PROGRAM :**

class MyClass {

    static int count = 0;

    final double PIE = 3.1415;

//CONSTRUCTORS

public MyClass() {

        count++;

    }

//MAIN FUNCTION

    public static void main(String[] args) {

        //CREATING OBJECTS

        MyClass obj1 = new MyClass();

        MyClass obj2 = new MyClass();

        MyClass obj3 = new MyClass();

        //PRINTING COUNT

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

        System.out.println("PIE constant value: " + obj1.PIE);

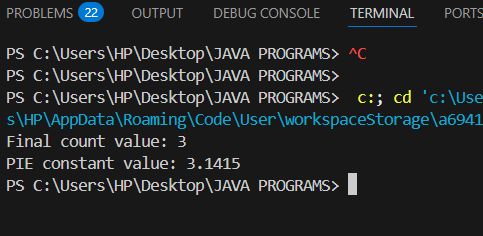
    }

}

**CLASS DIAGRAM :**

|  |
| --- |
| Myclass |
| Count : int  PIE : double |
| +Myclass() |

**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-5**

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

**Program :**

public class calculator {

    int a = 18;

    int b = 1;

void addition(){

    System.out.println("Addition is :" + (a+b));

}

}

class multi1 extends calculator{

    void subtraction(){

        System.out.println("Subtraction is :" + (a-b));

    }

}

class multi2 extends calculator{

    void multiplication(){

        System.out.println("Multiplication is :" + (a\*b));

    }

}

class multi3 extends calculator{

    void division(){

        System.out.println("Division is :" + (a/b));

    }

}

class multipleinheritance {

    public static void main(String[] args) {

        multi1 m1 = new multi1();

        multi2 m2 = new multi2();

        multi3 m3 = new multi3();

        m1.addition();

        System.out.println("    ");

        m1.subtraction();

        System.out.println("    ");

        m2.multiplication();

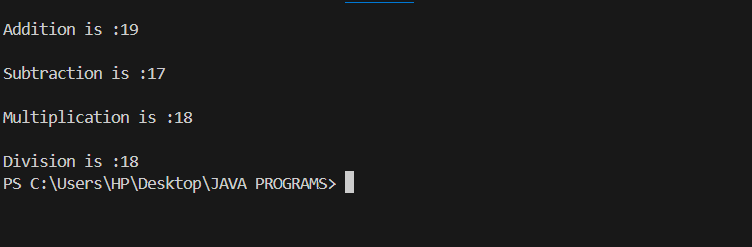
        System.out.println("    ");

        m3.division();

}

}

**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 :**

class Vehicle{

    String brand;

    int speed;

    Vehicle(String brand,int speed){

        this.brand=brand;

        this.speed=speed;

    }

    void Details(){

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

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

    }

}

class CARS extends Vehicle{

    int doors;

    int capacity;

    public CARS(String brand,int speed,int doors,int capacity){

        super(brand, speed);

        this.doors=doors;

        this.capacity=capacity;

    }

    void cardetails(){

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

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

    }

}

class Bikes extends Vehicle{

    Boolean gears;

    Bikes(String brand,int speed,Boolean gears){

        super(brand, speed);

        this.gears=gears;

    }

    void bikedetails(){

        if (gears==true)

        System.out.println("This bike has gears.");

        else

        System.out.println("This bike does not have gear system.");

    }

}

class Trucks extends Vehicle{

    int tons;

    Trucks(String brand,int speed,int tons){

        super(brand, speed);

        this.tons=tons;

    }

    void truckdetails(){

        System.out.println("The capacity of truck is: "+tons);

    }

}

class Rent{

    public static void main(String[] args){

        CARS c=new CARS("Tayota",120,5,5);

        c.cardetails();

        c.Details();

        Bikes b=new Bikes("KTM",80,true);

        b.bikedetails();

        b.Details();

        Trucks t=new Trucks("TATA",100,1);

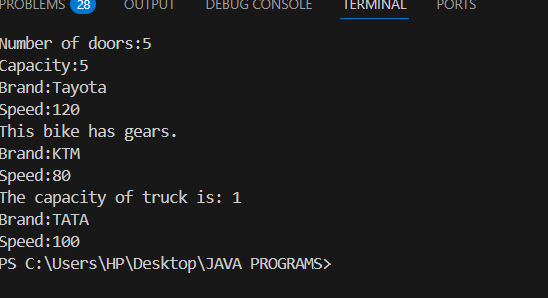
        t.truckdetails();

        t.Details();

    }

}

**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{

    String company;

    String model;

    String fuel;

    int capacity;

    void displayInfo(String company,String model,String fuel,int capacity){

        System.out.println("The details of vehicle: ");

        this.company=company;

        this.model=model;

        this.fuel=fuel;

        this.capacity=capacity;

    }

}

class car extends vehicle{

    void displayInfo(String company,String model,String fuel,int capacity){

        System.out.println("Company: "+company);

        System.out.println("Model: "+model);

        System.out.println("Fuel: "+fuel);

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

    }

}

class poly1{

    public static void main(String[] args){

        car car1=new car();

        car1.displayInfo("Audi","Model","Diesel",4);

    }

}

**OUTPUT :**

**ERRORS :**

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

**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 :**

class College{

    String name;

    int percentage;

    void geteligibility(String name,int percentage){

        this.name=name;

        this.percentage=percentage;

    }

}

class UG extends College{

    void geteligibility(String name,int percentage){

        if (percentage>=60){

            System.out.println(name+" is eligible");

        }

        else{

            System.out.println(name+" is not eligible");

        }

    }

}

class PG extends College{

    void geteligibility(String name,int percentage){

        if (percentage>=70){

            System.out.println(name+" is eligible");

        }

        else{

            System.out.println(name+" is not eligible");

        }

    }

}

class poly2{

    public static void main(String[] args){

        UG ug=new UG();

        ug.geteligibility("Person-1",40);

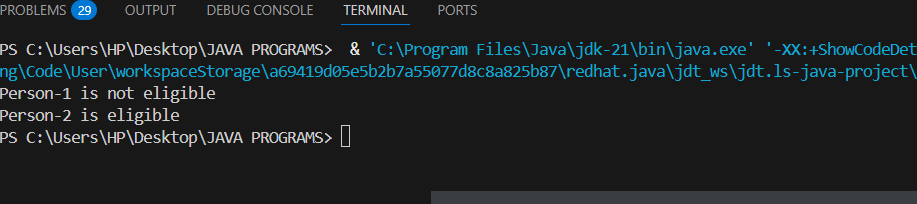
        PG pg=new PG();

        pg.geteligibility("Person-2",80);

    }

}

**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 Calcee{

    public int add(int a,int b){

        return a+b;

    }

    public double add(double a,double b){

        return a+b;

    }

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

        return a+b+c;

    }

}

class poly3{

    public static void main(String[] args){

        Calcee C1=new Calcee();

        System.out.println("Sum of 2 and 5 is: "+C1.add(2,5));

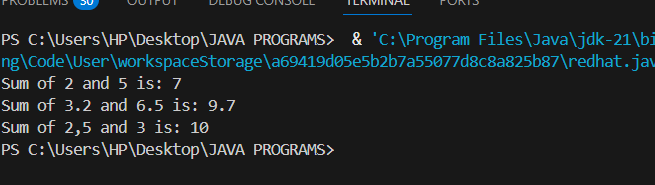
        System.out.println("Sum of 3.2 and 6.5 is: "+C1.add(3.2,6.5));

        System.out.println("Sum of 2,5 and 3 is: "+C1.add(2,5,3));

    }

}

**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 { // class shape

    void calculateArea( int a) { // method 1

        System.out.println("The area of Square is :" + (a\*a) );

    }

    void calculateArea(int a , int b) { // method 2

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

    }

}

class circle extends Shape { // inheritance class

    void calculateArea(double a){ // method overloading

        System.out.println("The area of circle is :" + (3.14\*a\*a));

  } }

class main { // main program

    public static void main(String[] args) {

        // creating objects for class

        Shape s = new Shape();

        circle c = new circle();

        // calling methods

        s.calculateArea(4);

        System.out.println("    ");

        s.calculateArea(4, 5);

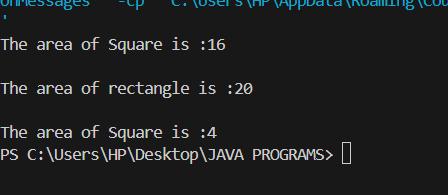
        System.out.println("    ");

        c.calculateArea(2);

    }

}

**OUTPUT :**

****

**ERRORS :**

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