Lab Manual

**Object oriented programming**

**23CSE101**



Department of computer science of Engineering

Amrita School of Computing

Amrita Vishwa Vidyapeetham,

Amaravathi campus

Name: G.J.N.S.Chaitanya

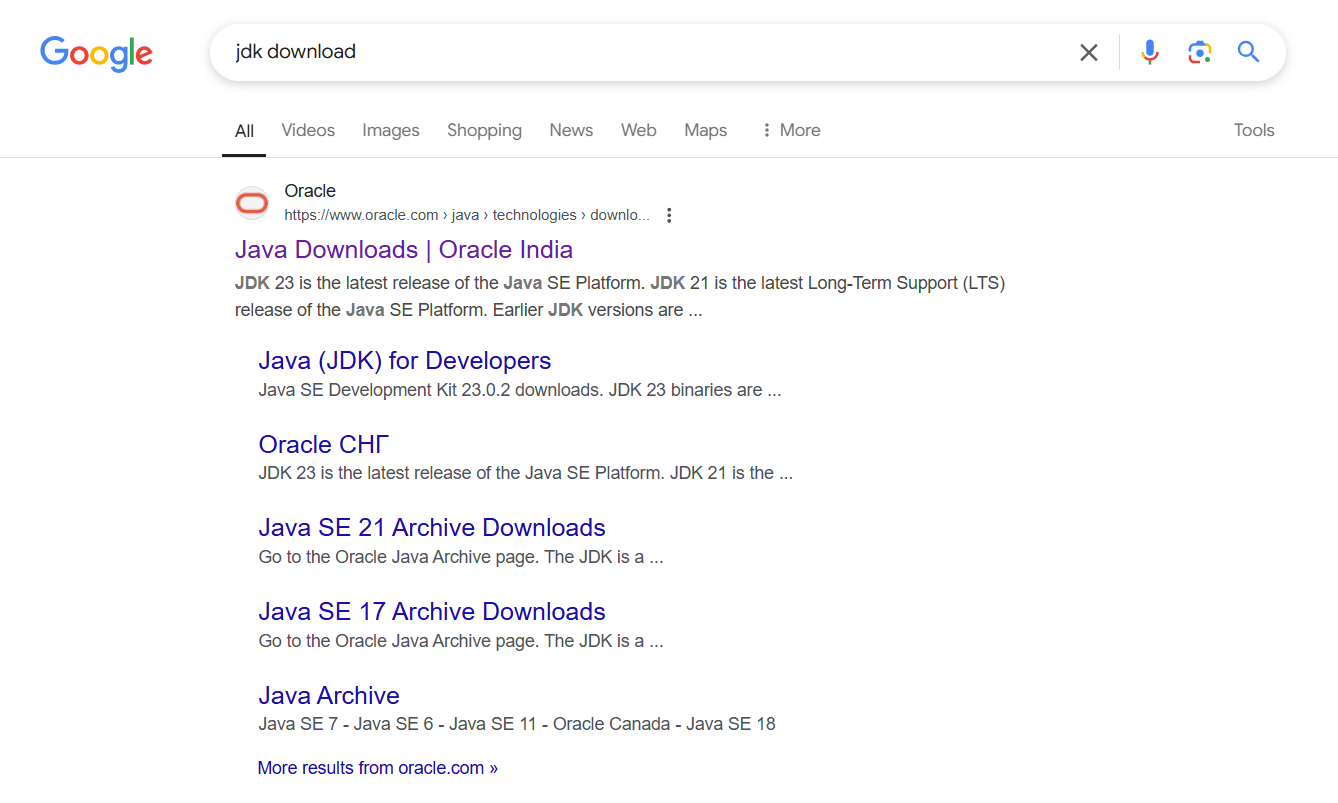
Roll no:AV.SC.U4CSE24115

WEEK-1

Aim: To download and install JAVA Compiler on our laptops.

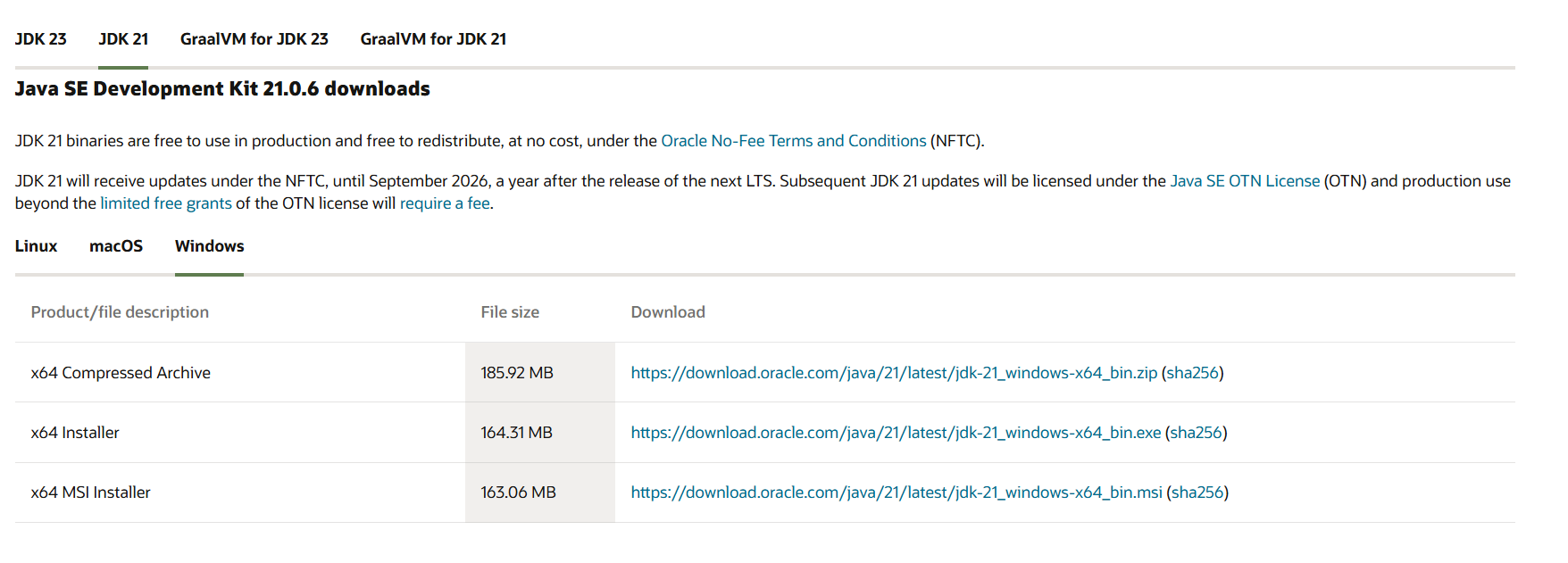
Procedure:

**Step1:** Search “JDK download” using a search engine



Step2: Click on the ORACEL website to download java

Step3: Select the 21.0.6 version which is suitable and stable for your system



Step4: Choose the appropriate operating system

Step5: Start downloading.

Step6: Once downloaded, navigate to the downloads folder and open the file

Step7: Click Next, accept the terms and conditions, and proceed with the installation

Step8: Java compiler will now be installed successfully.

Step9: After installation, open environment variables by searching for it on your laptop

A screenshot of a computer program

Description automatically generated

Step10: In environment variables, navigate to the system variables and select path option.

Step11: Click Edit, create a new path, and enter the required details.

A screenshot of a computer program

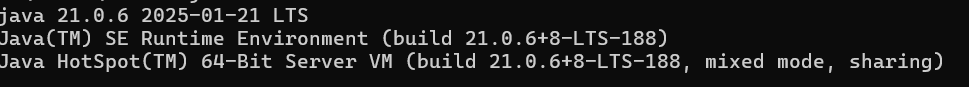
Description automatically generated

Step12: Copy the path, Click OK, return to System variables, and add a new variable named JAVA\_HOME.Paste the copied path and click OK to save.

A screenshot of a computer program

Description automatically generated

Step13: To verify the installation, open the command prompt and check the java version using: java –version.

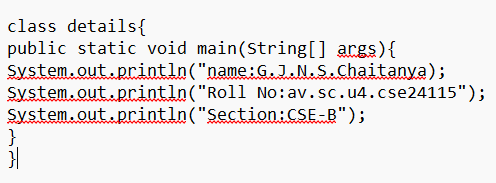


Step14: if the version is displayed, java has been successfully installed

AIM: To print the details of student using Java.

Procedure:

Step1: Open notepad and write the java code.



Step2: Save the file in a designated folder.

Step3: Open the command prompt.

Step4: Navigate to the file location and compile the code using: javac FILEname.java

A screenshot of a computer screen

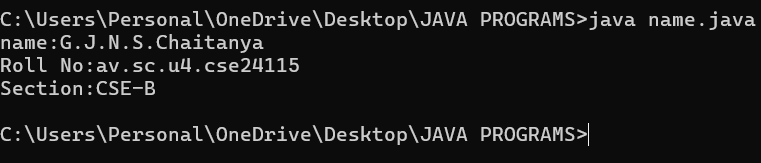
Description automatically generated

Step5: After successful compilation, a.class file will be generated

A screenshot of a computer

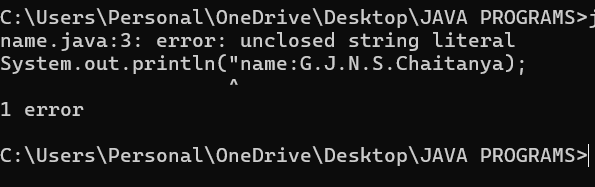
Description automatically generated

Step 6: To see the output type java and file name in command prompt by java FILE name



Step 7: The output will be displayed in the Command Prompt.

MY ERROR:



Ensure that after completion the statement close it with single or double quotes.

WEEK-2

1. AIM:

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

**User**

**Code:**



**Output:**

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

**2 )AIM:**

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

**input from user**

**code:**



**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** |



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

**3) AIM:**

**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);**

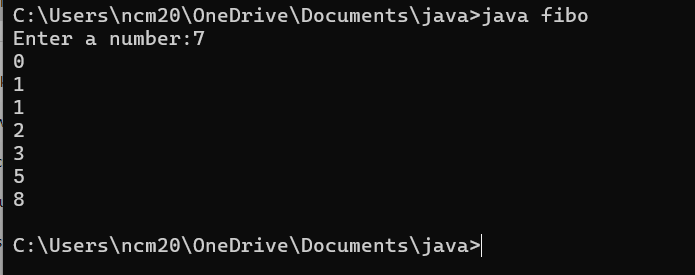
**i = i+1;**

**}**

**}**

**}**

**Output:**

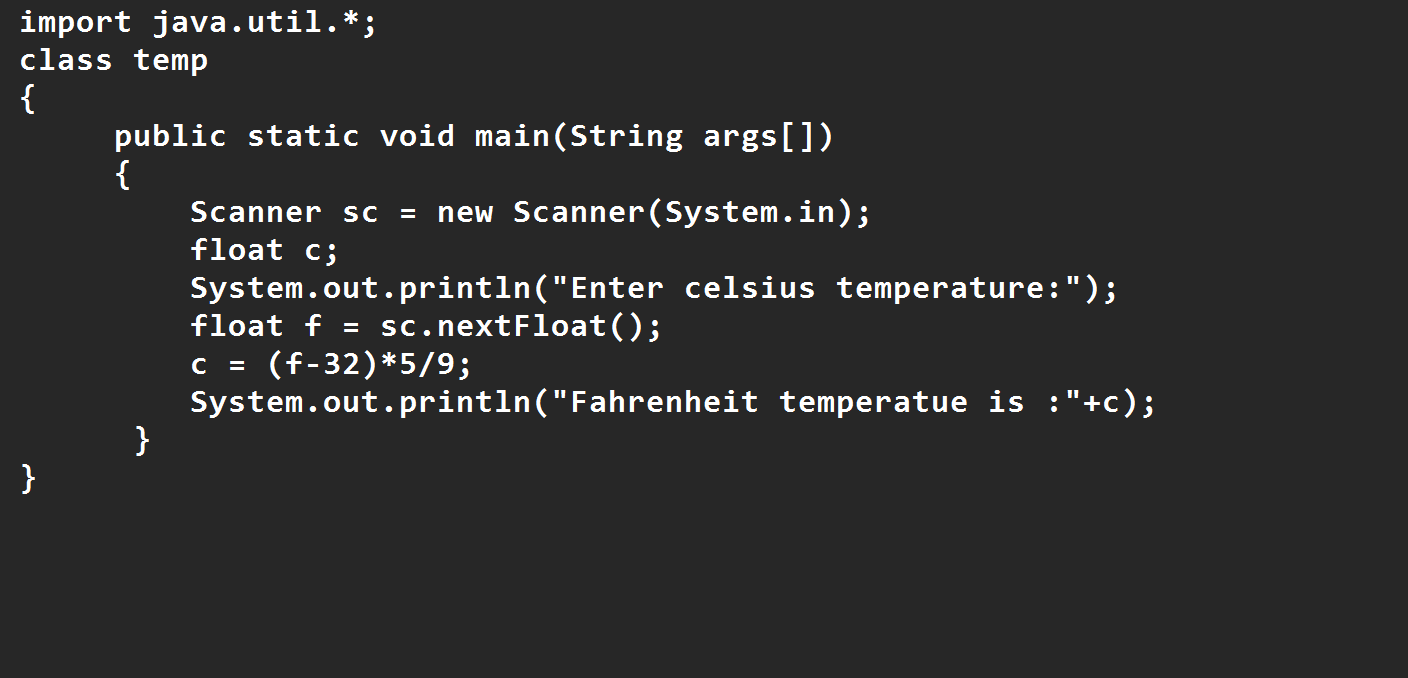


|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **Rectification** |
| **1** | **Logical error** | **Incorrect formula** | **Formula rectified** |
| **2** | **Run-time error** | **Incorrect path** | **Added correct path** |
| **3** |  |  |  |

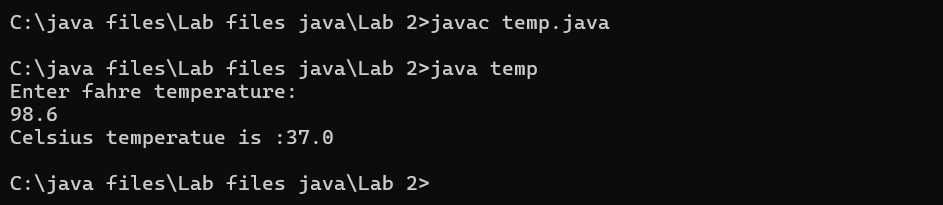
**5) AIM**

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

**Code:**



**Output:**



|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error type** | **Reason for error** | **rectification** |
| **1** | **Syntax error** | **Missing ”** | **“ is added** |
| **2** | **Missing import error** | **Util package missing** | **Util package added** |
| **3** |  |  |  |

**AIM:**

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

**Code**



**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** |
| **3** |  |  |  |

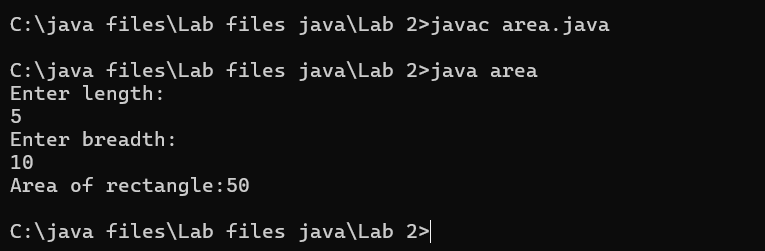
**6) AIM:**

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

**Code:**



**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** |  |  |  |

**7)AIM:**

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

**Code:**



**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**

**Aim:**

**To create java program with following instructions**

**1.Create a class with name car**

**2. Create four attributes named car\_color ,Car\_brand,fuel\_type,mileage**

**3. Create three methods named start(), stop(). Service()**

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

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

**class car**

**{**

**public String Car\_color;**

**public String Car\_brand;**

**public String fuel\_type;**

**public int mileage;**

**public void start()**

**{**

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

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

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

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

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

**}**

**public void service()**

**{**

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

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

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

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

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

**}**

**public void stop()**

**{**

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

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

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

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

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

**}**

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

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

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

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

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

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

**car1.mileage = 10;**

**car1.start();**

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

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

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

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

**car2.mileage = 300;**

**car2.stop();**

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

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

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

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

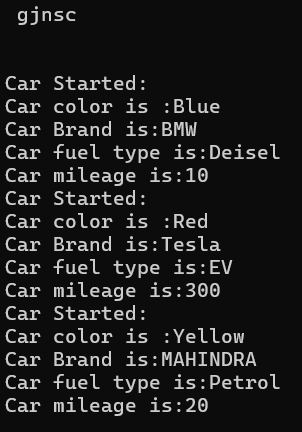
**car3.mileage = 20;**

**car3.service();**

**}**

**}**

**Output:**

****

**Class Diagram**

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

**2.AIM:**

**To create a class bankAccount with methods deposit() and withdrawl**

**Code:**

**class BankAccount {**

**private double balance;**

**public BankAccount(double initialBalance) {**

**if (initialBalance > 0) {**

**this.balance = initialBalance;**

**} else {**

**this.balance = 0;**

**}**

**}**

**public void deposit(double amount) {**

**if (amount > 0) {**

**balance = balance + amount;**

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

**} else {**

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

**}**

**}**

**public double getBalance() {**

**return balance;**

**}**

**}**

**public class Main {**

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

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

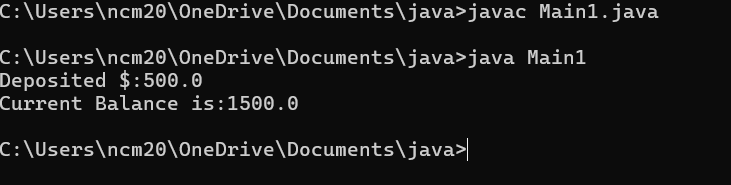
**account.deposit(500);**

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

**}**

**}**

**Output:**



**ERRORS:**

|  |  |  |
| --- | --- | --- |
| **Sno.** | **Error message** | **Error rectification** |
| **1.** | **error: ';' expected**  **cust1.withdraw(3050)** | **Add a “;”**    **cust1.withdraw(3050);** |

**WEEK-4**

**1.AIM:**

**WRITE A JAVA PROGRAM WITH CLASS NAMED “Book”. THE CLASS SHOUKD CONTAIN VARIOUS ATTRIBUTES SUCH AS TITLE, AUTHOR, YEAR OF PUBLICATION. IT SHOULD ALSO CONTAIN A CONSTRUCTOR WITH PARAMETERS WHICH INITIALIZES TITLE, AUTHOR, YEAR OF PUBLICATION AND CREATE A METHOD WHICH DISPLAYS THE DETAILS OF 2 BOOKS.**

**PROGRAM:**

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:



NEGATIVE CASE:

A black screen with white text

AI-generated content may be incorrect.

ERROR:

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

CLASS DIAGRAM:

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

**2.AIM:**

**WRITE A JAVA PROGRAM WITH CLASS NAMED “MyClass” WITH STATIC VARIABLE COUNT OF INT TYPE INTIALIZE IT TO ZERO AND CONSTANT “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 THEFINAL VALUES OF ‘COUNT’ AND ‘PI’ VARIABLES AND CREATE 3 OBJECTS.**

**PROGRAM:**

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();

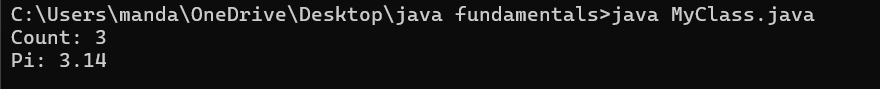
System.out.println("Count: " +count);

System.out.println("Pi: " +pi);

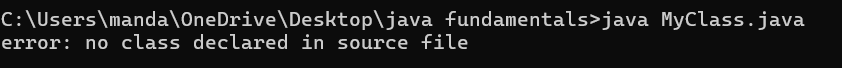
}

}

OUTPUT:



NEGATIVE CASE:



ERROR:

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

CLASS DIAGRAM:

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

WEEK-5

1) **Aim:** Create a calculator using the operations including addition, subtraction, multiplication and division using

multilevel in heritance and display the desired output.

- Important Points:

* + 1. Understand the calling of a Constructor
    2. Giving class name correctly
    3. Give the parameters Correctly

**Program:**

class easy{

void add(int a,int b){

System.out.println("Sum of Numbers is: "+(a+b));

}

void subtract(int a,int b){

System.out.println("Difference of 2 Numbers: "+(a-b));

}

}

class hard extends easy{

void product(int a,int b){

System.out.println("Product of 2 numbrs is: "+(a\*b));

}

}

class ultra extends hard{

void divide(int a,int b){

if (b!=0){

System.out.println("Dividing of 2 numbers is: "+(a/b));

}

else{

System.out.println("Denominator must not be zero");

}

}

}

class Calc{

public static void main(String[] args){

ultra d=new ultra();

d.add(6,9);

d.subtract(9,6);

d.product(23,3);

d.divide(4,2);

}

}

**OUTPUT:**



**Errors:**

|  |  |  |
| --- | --- | --- |
| S.NO | Error Name | Error Rectification |
| 1 | Syntax/ Compilation Error | Absence of Semicolon |
| 2 | Closing Brackets | Need to Close the brackets |
| 3 | Class Name Error | Give the class name correctly |
| 4 | Constructor Calling | Call the constructor correctly |

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

1. Cars should have an additional property: number of doors, Seating capacity.
2. Bikes should have a property indicating whether they have gears or not. iii. The system should also include a function to display details about each vehicle and indicate when a vehicle is starting. iv. Each class should have a constructor.

Questions:

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?

1. Truck should include and additional property capacity (in tons).
2. Create a showTruck() method to display the truck’s capacity.
3. Write a constructor for truck that initializes all properties.

3. Implement the truck class and update the main method to create a Truck object and also create an object for car and bike subclasses. Finally display the details.

**- Important Points:**

1.Understand the calling of a Constructor

2.Giving class name correctly 3.Give the parameters Correctly

**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);

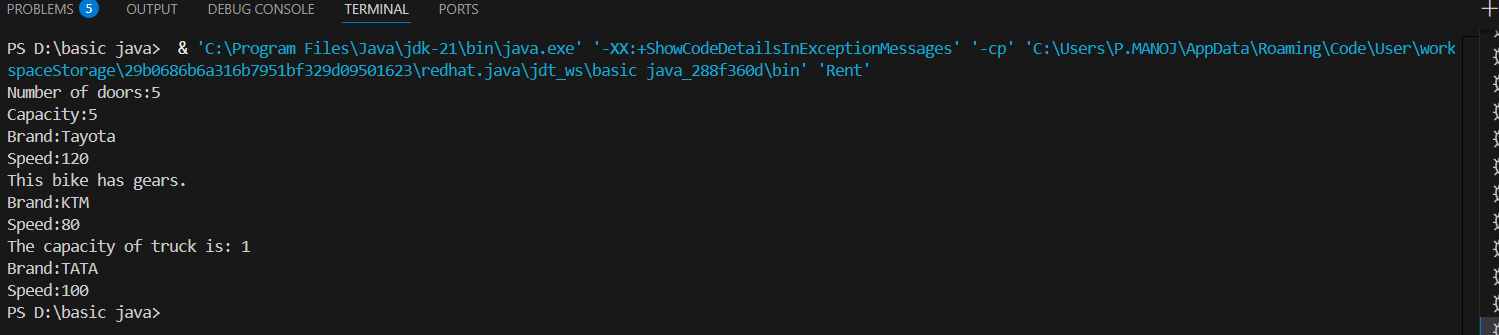
        t.truckdetails();

        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 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 cardetails{

    public static void main(String[] args){

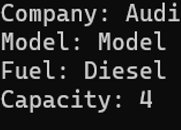
        car car1=new car();

        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);

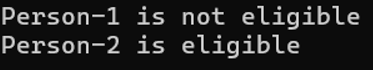
        PG pg=new PG();

        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));

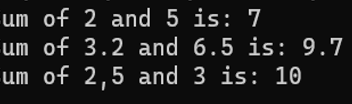
        System.out.println("Sum of 3.2 and 6.5 is: "+C1.add(3.2,6.5));

        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) {

  Shape s = new Shape();

        circle c = new circle();

      s.calculateArea(4);

        System.out.println("    ");

        s.calculateArea(4, 5);

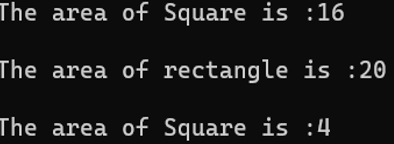
        System.out.println("    ");

        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 |

**WEEK-7**

**1) Aim : Write a Java program to create an abstract class Animal with an abstract method called sound(). Create subclasses Lion and Tiger that extend the Animal class and implement the sound() method to make a specific sound for each animal.**

**Important points :**

1. **Here we used abstract class concept it is a restricted class that cannot be instantiated (cannot have objects created directly) and is typically designed to be extended by subclasses.**
2. **An abstract method is a method declared in an abstract class that does not have an implementation, meaning it doesn't have a body within the abstract class.**

**Here we used hierarchy inheritance concept means every sub class extends super class.**

**Program :**

abstract class animal{

abstract void sound();}

class lion extends animal{

void sound(){

System.out.println("lion roars"); } }

class tiger extends animal{

void sound(){

System.out.println("tiger growls"); }}

class p17{

public static void main(String[] args) {

System.out.println("NAME : G.CHAITANYA;ROLL NO : av.sc.u4CSE24115;SEC: CSE-B");

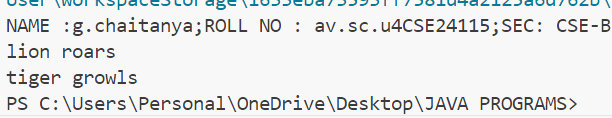
lion l = new lion();

l.sound();

tiger t = new tiger();

t.sound(); } }

**Output**

****

**Error table**

|  |  |  |
| --- | --- | --- |
| **Error table :** S.NO | Error Name | Error Rectification |
| **1** | Main Class | Better to create main class name same as the file you saved and first letter is capital. |
| **2** | Method | We need to provide return type to the method. |
| **3** | Data type | As per need provide data type |
| **4** | Abstract method | Implementation in subclass only |

1. Aim : Write a Java program to create an abstract class Shape3D with abstract methods calculateVolume() and calculateSurfaceArea(). Create subclasses Sphere and Cube that extend the Shape3D class and implement the respective methods to calculate the volume and surface area of each shape.

Important points :

1. Here we used abstract class concept it is a restricted class that cannot be instantiated (cannot have objects created directly) and is typically designed to be extended by subclasses.

2. An abstract method is a method declared in an abstract class that does not have an implementation, meaning it doesn't have a body within the abstract class.

3. Understanding the calling of constructor.

**Program :**

abstract class Shape3D {

abstract double calculateVolume();

abstract double calculateSurfaceArea();

}

class Sphere extends Shape3D {

int radius;

Sphere(int radius) {

this.radius = radius;

}

double calculateVolume() {

return (4.0 / 3.0) \* Math.PI \* Math.pow(radius, 3);

}

double calculateSurfaceArea() {

return 4 \* Math.PI \* Math.pow(radius, 2);

}

}

class Cube extends Shape3D {

int edge;

Cube(int edge) {

this.edge = edge;

}

double calculateVolume() {

return Math.pow(edge, 3);

}

double calculateSurfaceArea() {

return 6 \* Math.pow(edge, 2);

}

}

public class Week7{

public static void main(String[] args) {

Shape3D s = new Sphere(5);

System.out.println("Sphere Volume: " + s.calculateVolume());

System.out.println("Sphere Surface Area: " + s.calculateSurfaceArea());

Shape3D c = new Cube(6);

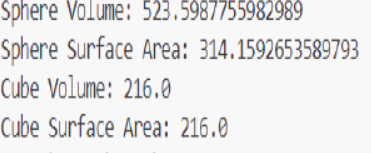
System.out.println("Cube Volume: " + c.calculateVolume());

System.out.println("Cube Surface Area: " + c.calculateSurfaceArea());

}

}

**Output :**



**Error table**

|  |  |  |
| --- | --- | --- |
| S.NO | Error Name | Error Rectification |
| **1** | Main Class | Better to create main class name same as the file you saved and first letter is capital. |
| **2** | Data type | As per need provide data type |
| **3** | Abstract method | Implementation in subclass only |

* 1. 3) **Aim :** write a java program using an abstract class to define a method for pattern printing Here we used nested for loop concept the block of code is executed until the condition is false.
  2. Here the logic very important .
  3. Here we used abstract class concept it is a restricted class that cannot be instantiated (cannot have objects created directly) and is typically designed to be extended by subclasses.

Create an abstract class named pattern printer with an abstract method print pattern (int n)

and a concrete method to display the pattern tittle.

Implement two sub class :

1.star pattern -prints a right-angled triangle of stars(\*)

2.number pattern-prints a right angled triangle of increasing numbers.

In the main () method,create objects of both subclasses and print the pattern for a given number of rows.

**Important points :**

**Program :**

abstract class PatternPrinter{

public abstract void printPattern(int n);

public void printTitle(String title) {

System.out.println(title);

}

}

class StarPattern extends PatternPrinter {

@Override

public void printPattern(int n){

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

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

System.out.print("\* ");

}

System.out.println();

}

}

}

class NumberPattern extends PatternPrinter {

@Override

public void printPattern(int n) {

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

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

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

}

System.out.println();

}

}

}

public class Week7\_P8 {

public static void main(String[] args) {

int rows=5;

PatternPrinter starPattern = new StarPattern();

PatternPrinter numberPattern = new NumberPattern();

starPattern.printTitle("Star Pattern:");

starPattern.printPattern(rows);

System.out.println();

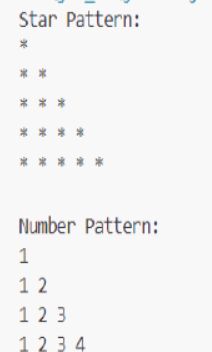
numberPattern.printTitle("Number Pattern:");

numberPattern.printPattern(rows);

}

}

**Output :**



**Error table:**

|  |  |  |
| --- | --- | --- |
| S.NO | Error Name | Error Rectification |
| **1** | Main Class | Better to create main class name same as the file you saved and first letter is capital. |
| **2** | Data type | As per need provide data type |
| **3** | Syntax in for | Initializing value and condition should be correct |
| **4** | overridding | Same method names |