

**(23CSE111) OBJECT ORIENTED PROGRAMMING**

**LAB MANUAL**

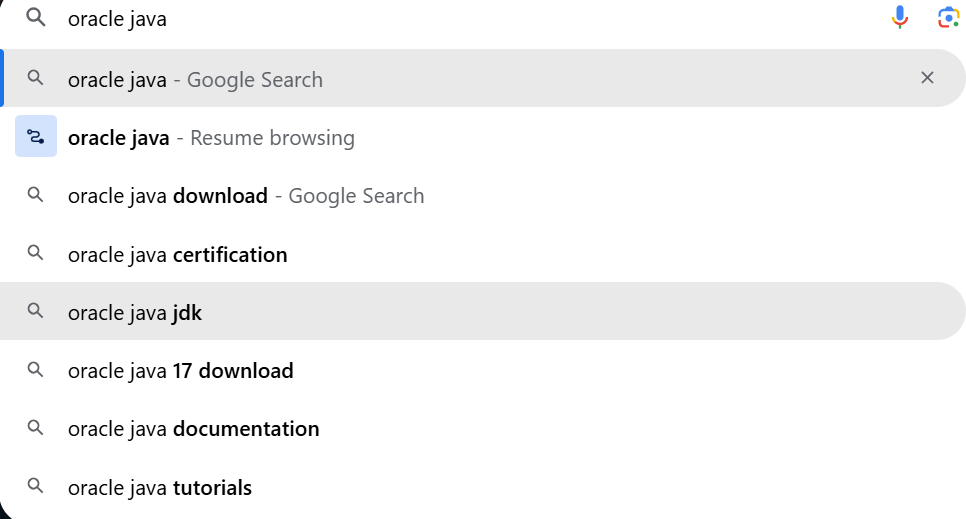
**CSE-1st YEAR II SEMESTER (2025-2026)**

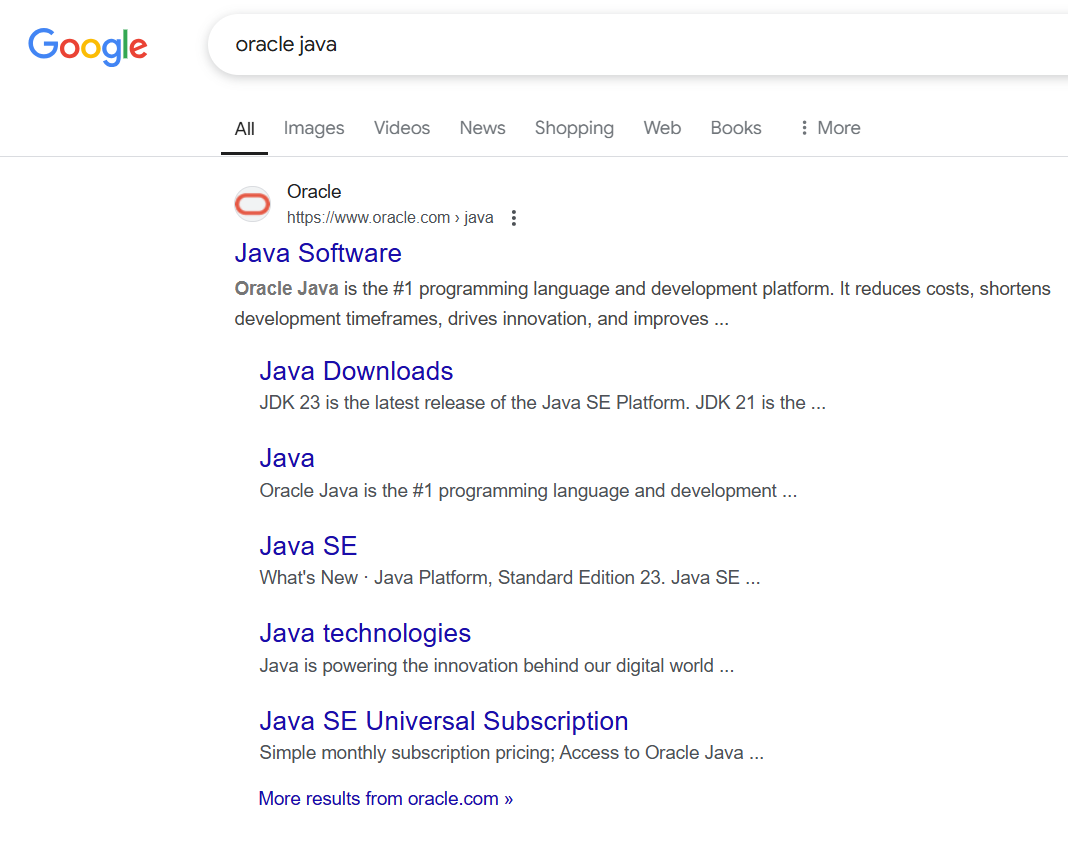
|  |  |  |  |
| --- | --- | --- | --- |
| **Submitted by** | | **Submitted to** | |
| **NAME:** | **Hari siva sai rohit.Thota** | **NAME:** | **RAJ KUMAR BATCHU** |
| **ROLL NO:** | **AV.SC.U4CSE24126** | **DEPARTMENT:** | **CSE** |
| **SECTION:** | **CSE-B** | **DESIGNATION:** | **Asst.Professor** |

|  |  |
| --- | --- |
| **MARKS:** |  |
| **SIGNATURE:** |  |
| **DATE:** |  |

WEEK-1

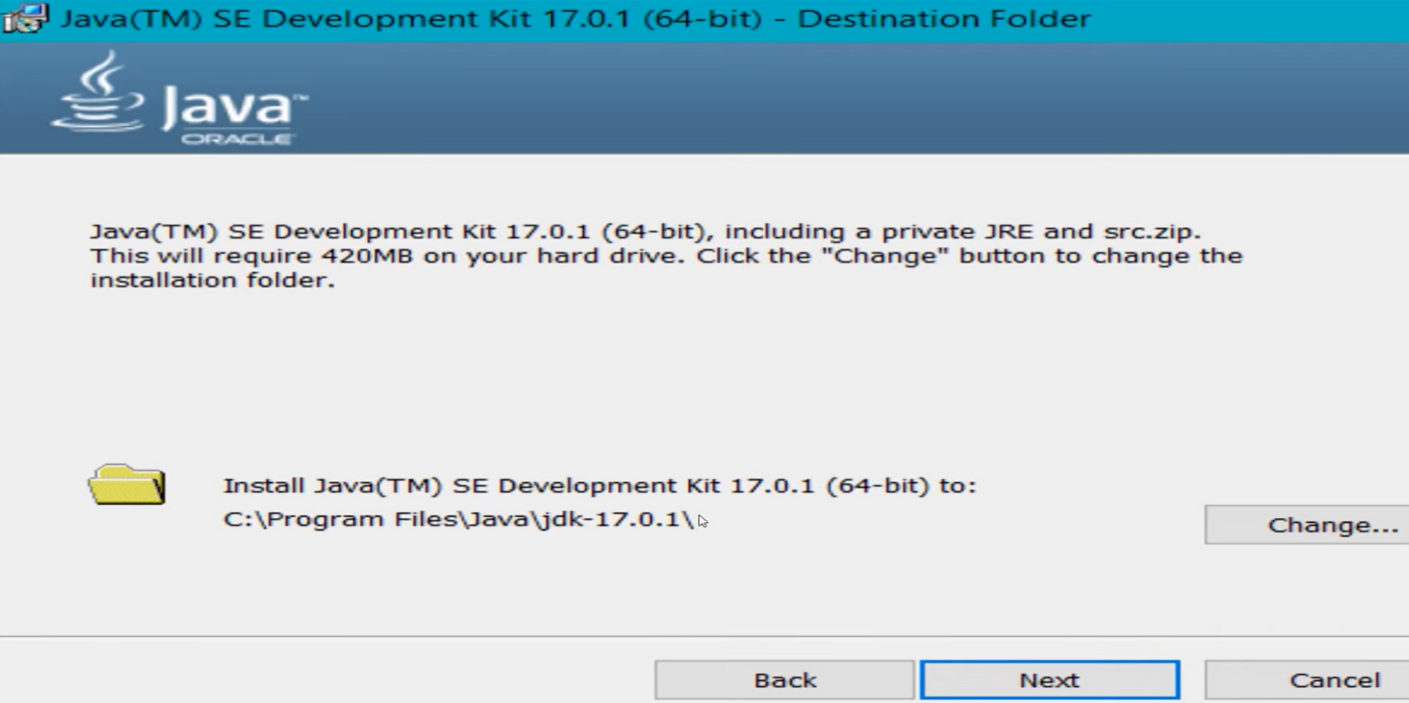
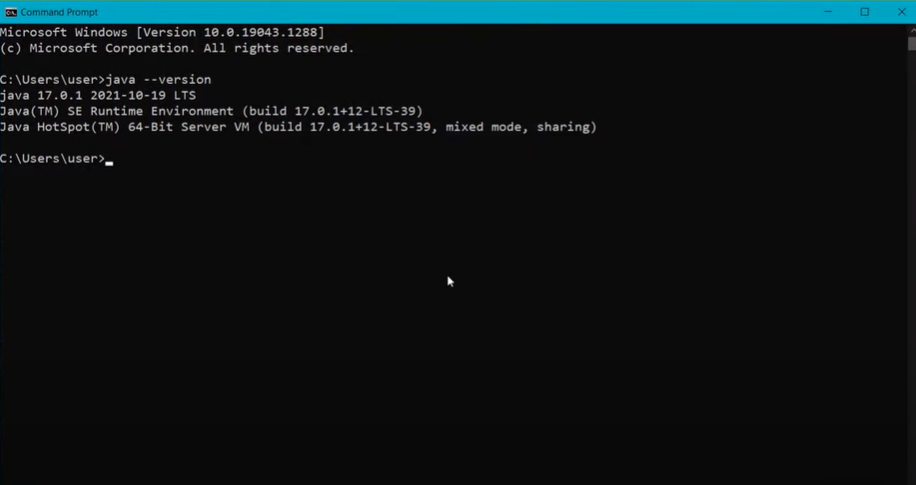
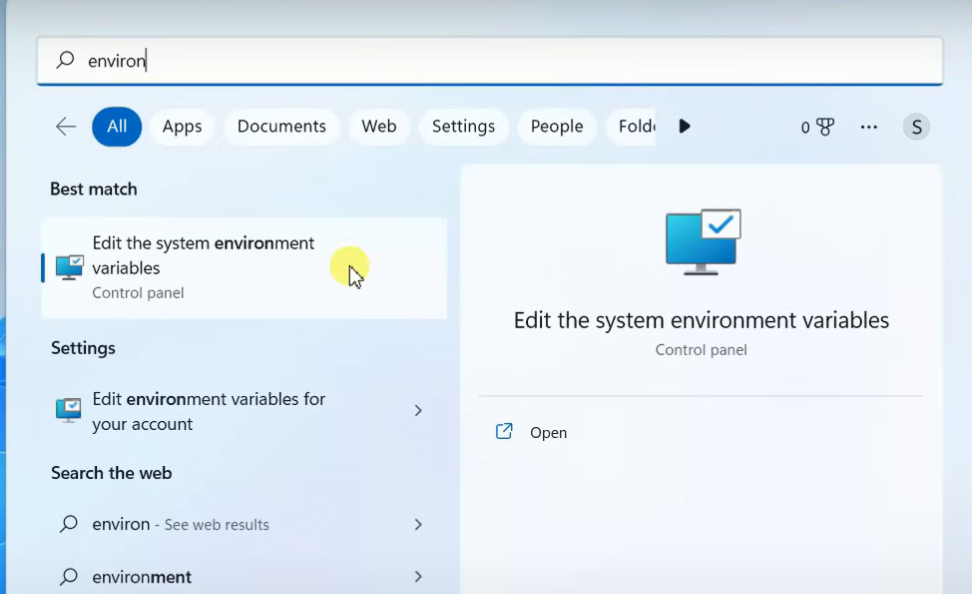
1)Aim:Explain the process of installing JDK (JAVA DEVELOPMENT KIT).

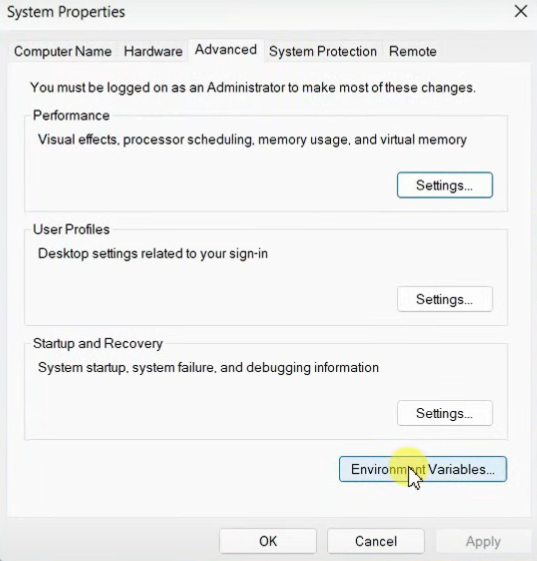
* Open “Oracle” in your web browser.
* Click on “ download Java ” in the oracle interface.
* Select the version you want to download (version JDK21 is best).
* Select the Operating System (OS) of your PC and click on the link “x64 installer” to download .
* **Installation:**
* **Then click on java downloads**

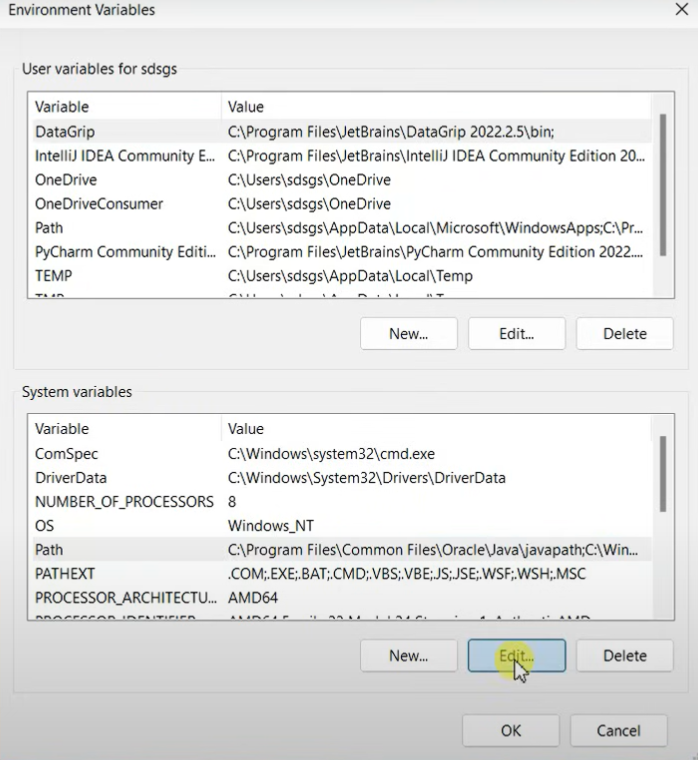
****

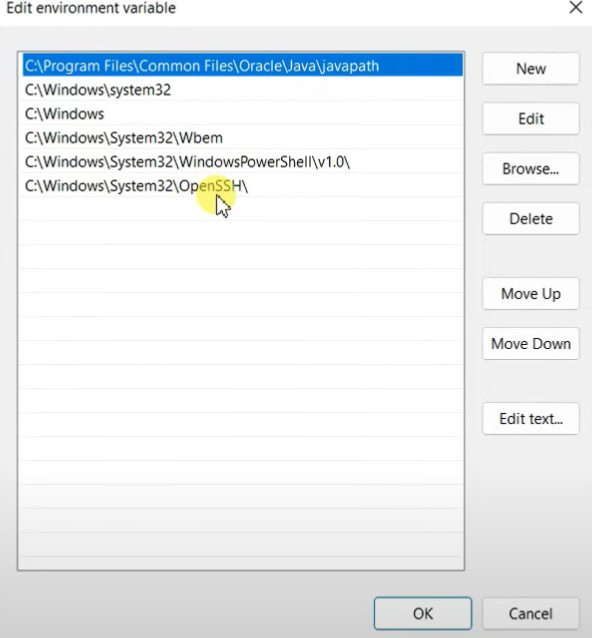
* **Click on next**

****

* ****Click on next
* **Click on close**
* To check whether java is downloaded correctly or not.
* Open command window and type “java --version**” .**
* To set the path:
* Go to “start” and se “Edit the search environment variables”.
* ****Click on open and continue to click on Environment variables.

****

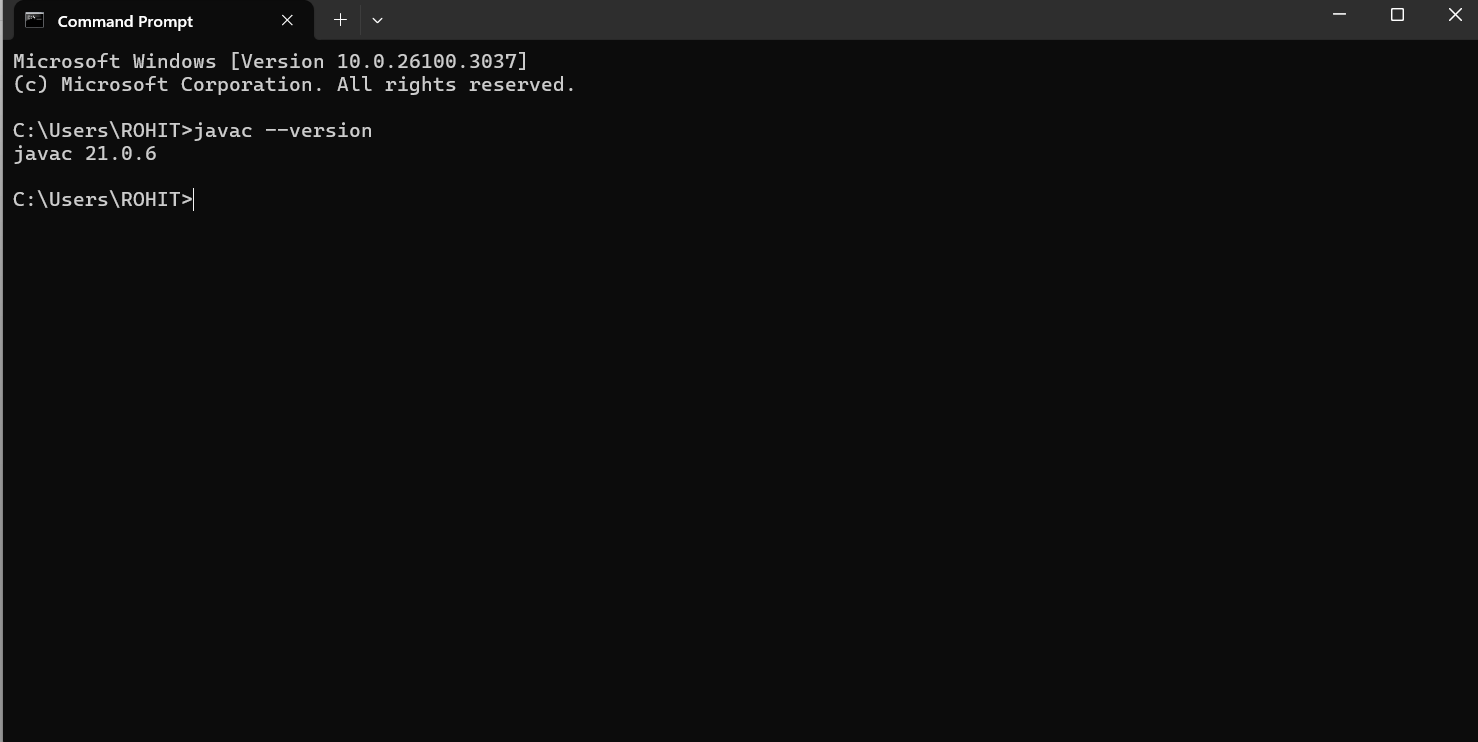
* Select the “path” option in “system variables”. 
* **Now click on new paste the path of java**

****

* To check whether the path is properly set or not

Open command window and type “javac --version”

If it is showing the exact version of java you are installed then the path is set properly.

****

* Now create a text document with “.java” extension and open in notepad ,

To write code.

**2) Aim: Write your code in notepad and execute in command prompt.**

**Code:**

class Student\_id

{

public static void main(String[] args) {

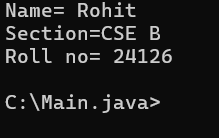
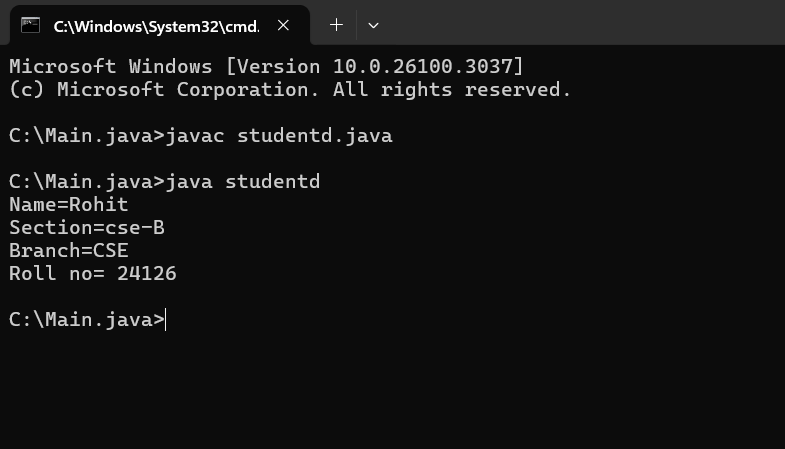
System.out.println("Name= Rohit");

System.out.println("Section=CSE B");

System.out.println("Roll no= 24126")

}

}

**** **OUTPUT:**

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Error** | **Rectification** |
| **1)** | **error: ‘;’ expected**  System.out.println("Roll no= 24126")  **^** | **In the code,semicolon must be added at the end of line.** |

**IMPORTANT POINTS:**

**1)System.out.println(“this string will be printed”) – this line of code is used to print any string.**

**2)if you want to save your java file as MyFirst then “class MyFirst { “ should be written”}**

WEEK-2

1)Aim: Write a java program for SI?

**import java.util.Scanner;**

**class largest\_number{**

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

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

**System.out.println("enter the first number:");**

**int a=l.nextInt();**

**System.out.println("enter the second number:");**

**int b=l.nextInt();**

**System.out.println("enter the third number:");**

**int c=l.nextInt();**

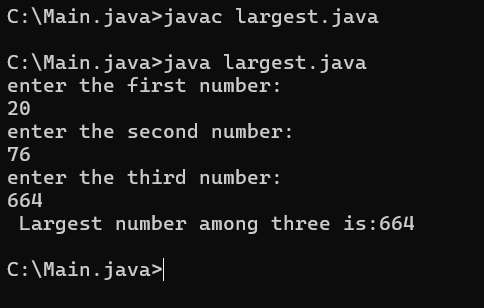
**int largest\_number = ((a > b && a > c)? a: (b > a && b>c)?b :c);**

**System.out.println(" Largest number among three is:" + largest\_number);**

**}**

**}**

**Output:**

****

|  |  |  |
| --- | --- | --- |
| S.no | error | rectification |
| 1) | scanner | “s” must be in capital letter. |

2)Aim: *Write a program in java for factorial of a number.*

**import java.util.Scanner;**

**class factorial {**

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

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

**System.out.println("enter the number: ");**

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

**long factorial = 1;**

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

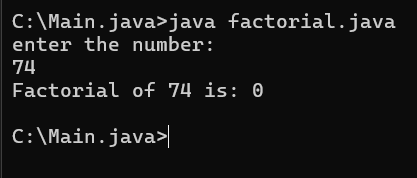
**factorial \*= i;**

**}**

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

**}**

**}**

**Output:**

|  |  |  |
| --- | --- | --- |
| code | error | rectification |
|  | For root in java we  don’t use \*\*. | Use Math.sqrt  statement.  Double  area=Math.sqrt(x); |

**3)Aim:**Write a program in java for fibonacci series.

**import java.util.Scanner;**

**class Fibonacci {**

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

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

**System.out.println("enter the number: ");**

**int n=f.nextInt();**

**int firstTerm = 0, secondTerm = 1;**

**System.out.println("Fibonacci Series up to " + n + " numbers:");**

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

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

**int nextTerm = firstTerm + secondTerm;**

**firstTerm = secondTerm;**

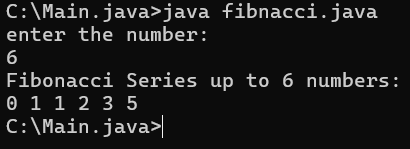
**secondTerm = nextTerm;**

**}**

**}**

**}**

**Output:**

****

|  |  |  |
| --- | --- | --- |
| S.no | error | rectification |
| 1) | For increment we have used +i | We must use ++i for increment |

**4)a)**Write a program in java for converting temperature from celsius to fahrenite***.***

**import java.util.Scanner;**

**class temperature\_conversion{**

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

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

**System.out.println("enter the Celcius :");**

**float ctemp=input.nextFloat();**

**float ftemp;**

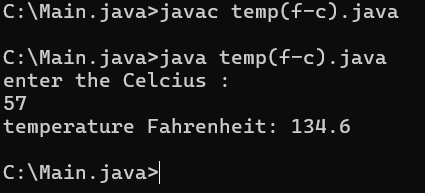
**ftemp=(ctemp\*9/5)+32;**

**System.out.println("temperature Fahrenheit: " + ftemp);**

**}**

**}**

**Output:**

****

|  |  |  |
| --- | --- | --- |
| code | error | rectification |
|  | System.out.println(“temperature Fahrenhiet: +ftemp) | System.out.println(“temperature  Fahrenhiet: “+ftemp) |

**4)b)**Write a program in java for converting temperature from fahrenite to celsius.

**import java.util.Scanner;**

**class temperature\_conver{**

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

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

**System.out.println("enter the temperature in Fahrenheit:");**

**float ftemp=input.nextFloat();**

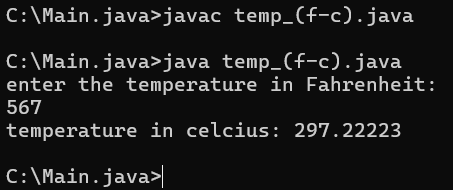
**float ctemp;**

**ctemp=(ftemp-32)\*5/9;**

**System.out.println("temperature in celcius: " +ctemp);**

**}**

**}**

**Output:**

|  |  |  |
| --- | --- | --- |
| **S.no** | **error** | **Rectification** |
| **1)** | **Strings**  **System.out.println("temperature in celcius: + ctemp);** | **System.out.println("temperature in celcius:” + ctemp);** |

**5)Aim:** Write a program in java for area of rectangle

**import java .util. Scanner;**

**class area\_rectangle{**

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

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

**System.out.println("enter the length:");**

**float l=a.nextFloat();**

**System.out.println("enter the width:");**

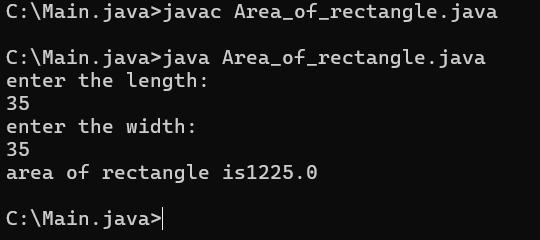
**float w=a.nextFloat();**

**float area=l\*w;**

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

**}**

**}**

**Output:**

|  |  |  |
| --- | --- | --- |
| code | error | rectification |
|  | Float area=l\*\*w; | For multiplication we should use “ \* ” for one time,so  Float area=l\*w; |

**6)Aim:** Write a java program for SI?

**import java.util.Scanner;**

**class SI{**

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

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

**System.out.print("enter the principal amount");**

**float p=input.nextFloat();**

**System.out.print("rate of the intrest ");**

**float R=input.nextFloat();**

**System.out.print("enter time in years");**

**float T=input.nextFloat();**

**float SI;**

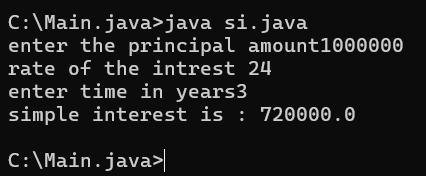
**SI=(p\*T\*R)/100;**

**System.out.println("simple interest is : "+SI);**

**}**

**}**

**Output:**

****

|  |  |  |
| --- | --- | --- |
| S.no | error | rectification |
| 1) | SI=P\*T\*R/100 | SI=( P\*T\*R)/100 |

**7)Aim:** *Write a program in java for area of triangle using heron’s formula.*

**import java.util.Scanner;**

**class heronsformula{**

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

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

**System.out.println("enter the value for a :");**

**Double a=input.nextDouble();**

**System.out.println("enter the value for b :");**

**Double b=input.nextDouble();**

**System.out.println("enter the value for c :");**

**Double c=input.nextDouble();**

**Double s=(a+b+c)/2;**

**System.out.println("S is the value of semi perimeter"+s);**

**Double x=s\*(s-a)\*(s-b)\*(s-c);**

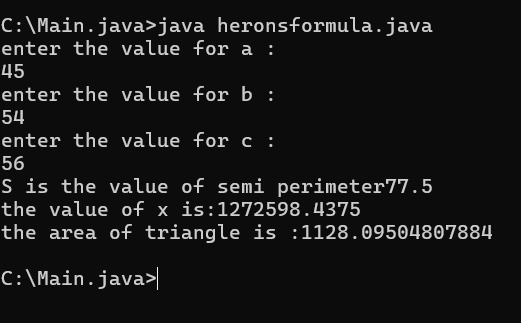
**System.out.println("the value of x is:"+x);**

**Double area=Math.sqrt(x);**

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

**}**

**Output:**

****

|  |  |  |
| --- | --- | --- |
| code | error | rectification |
|  | In formula ,  X=s\*(s-a)\*(s-b)\*(s\_c) | Instead of “ – “ we used “\_” there fore we got an error  X=s\*(s-a)\*(s-b)\*(s-c) |

**Week-3**

1. **Aim:Create the java program with the following instructions**
2. **Create a class with name Car**
3. **Create 4 attributes named Car\_Color , Car\_brand, fuel\_type, mileage**
4. **Create 3 method named Start( ) , Stop( ),  Service( )**
5. **Create 3 objects Car1 ,  Car2 , Car3**
6. **Create a constructor which should print “Welcome to Car Garage”**

**Code: class Car{**

**public String carColor;**

**private String carBrand;**

**private String fuelType;**

**public int mileage;**

**Car(String carColor , String carBrand , String fuelType , int mileage){**

**this.carColor = carColor;**

**this.carBrand = carBrand;**

**this.fuelType = fuelType;**

**this.mileage = mileage;**

**System.out.println(carColor + " " + carBrand + " " + fuelType + " " + mileage);**

**}**

**public void Start(){**

**System.out.println("The car has just started");**

**}**

**public void Stop(){**

**System.out.println("The car has just stopped");**

**}**

**public void Service(){**

**System.out.println("The car is in good condition");**

**}**

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

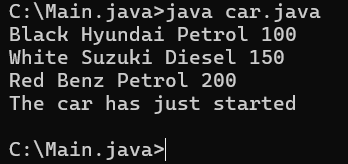
**Car Car1 = new Car("Black","Hyundai","Petrol",100);**

**Car Car2 = new Car("White","Suzuki","Diesel",150);**

**Car Car3 = new Car("Red","Benz","Petrol",200);**

**Car1.Start();**

**}}**

**Output:**

**Error:**

|  |  |  |
| --- | --- | --- |
| **s.no** | **Expected Error** | **Reason** |
| **1** | **}** | **} is expected at end of the calass** |
| **2** | **Setting the parameters inside  the constructer** | **Without setting the constructor we cannot pass the values** |

**Class Daigram:**

|  |
| --- |
| **Car** |
| **+ carColor : String**  **- carBrand : String**  **- fuelType : String**  **+ mileage : int** |
| **+ Car( ) : void**  **+ Start( ) : void**  **+ Stop( ) : void**  **+ Service( ) : void** |

**Important points:**

* **Private is an access specifier , It means attributes can be used in particular class only.**
* **Method is a part of code which only runs when it is called.**

**2)Aim:** **Write a java program to create a class BackAccount with two methods deposit( ) and withdraw( )**

1. **In deposit( ) whenever an amount is deposited it has to be updated with current amount**
2. **In withdraw( ) whenever an amount is withdrawn it has to be less than current amount else print “Insufficient funds”.**

**class BankAccount{**

**private String Name;**

**private int AccNo, CurrBal ;**

**BankAccount(String Name, int AccNo, int CurrBal){**

**this.Name = Name;**

**this.AccNo = AccNo;**

**this.CurrBal = CurrBal;**

**System.out.println("The customers are : " + this.Name + " ");**

**}**

**public int deposit(int dAmt){**

**CurrBal = CurrBal + dAmt ;**

**return CurrBal;**

**}**

**public void withdraw(int wAmount){**

**if(wAmount < CurrBal){**

**CurrBal = CurrBal - wAmount ;**

**System.out.println(CurrBal);**

**}**

**else{**

**System.out.println("Insufficient funds");**

**} }**

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

**BankAccount rohit = new BankAccount("hari",1500,10000);**

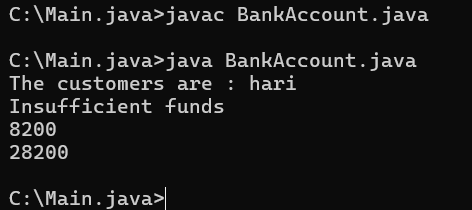
**rohit.withdraw(25000);**

**rohit.withdraw(1800);**

**int FinalAmount =rohit.deposit(20000);**

**System.out.println(FinalAmount);**

**} }**

**Output:**

**Errors :**

|  |  |  |
| --- | --- | --- |
| **s.no** | **Expected Errors** | **Reason** |
| **1** | **Giving the parameters inside the constuctor** | **We cannot pass the values inside the constructor without setting first** |
| **2** | **}** | **} is sometimes missing at the end of class** |

**Class Diagram :**

|  |
| --- |
| **BankAccount** |
| **-Name : String**  **- AccNo :String**  **-CurrlBal :String** |
| **+ Bank Account() :  void**  **+deposit() :int**  **+withdraw() : void** |

**Important points:**

* The constructor is called when an object of a class is created. It can be used to set initial values for object attributes.
* The name of the constructor must match with the name of the class.
* The constructor doesn’t have a return type.

**Week-4**

**1)Aim:** To create a java program with class named Myclass with a static variable “Count” of “int type”, Initialized to 0 and a constant variable “pi” of type double initialized to 3.1415 as attributes of that class Now, define a constructor for “Myclass” that increments the “Count” variable each that an object of Myclass is created. Finally , print the final values of “Count” and “pi” variables.

CODE:**class myclass{**

**static int count=0;**

**final double pi=3.1415;**

**void myclass(){**

**count=count+1;**

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

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

**}}**

**class details{**

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

**myclass obj1=new myclass();**

**myclass obj2=new myclass();**

**myclass obj3=new myclass();**

**myclass obj4=new myclass();**

**myclass obj5=new myclass();**

**obj1.myclass();**

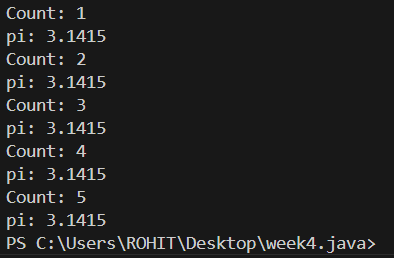
**obj2.myclass();**

**obj3.myclass();**

**obj4.myclass();**

**obj5.myclass();**

**}}**

**Output:**

**Errors :**

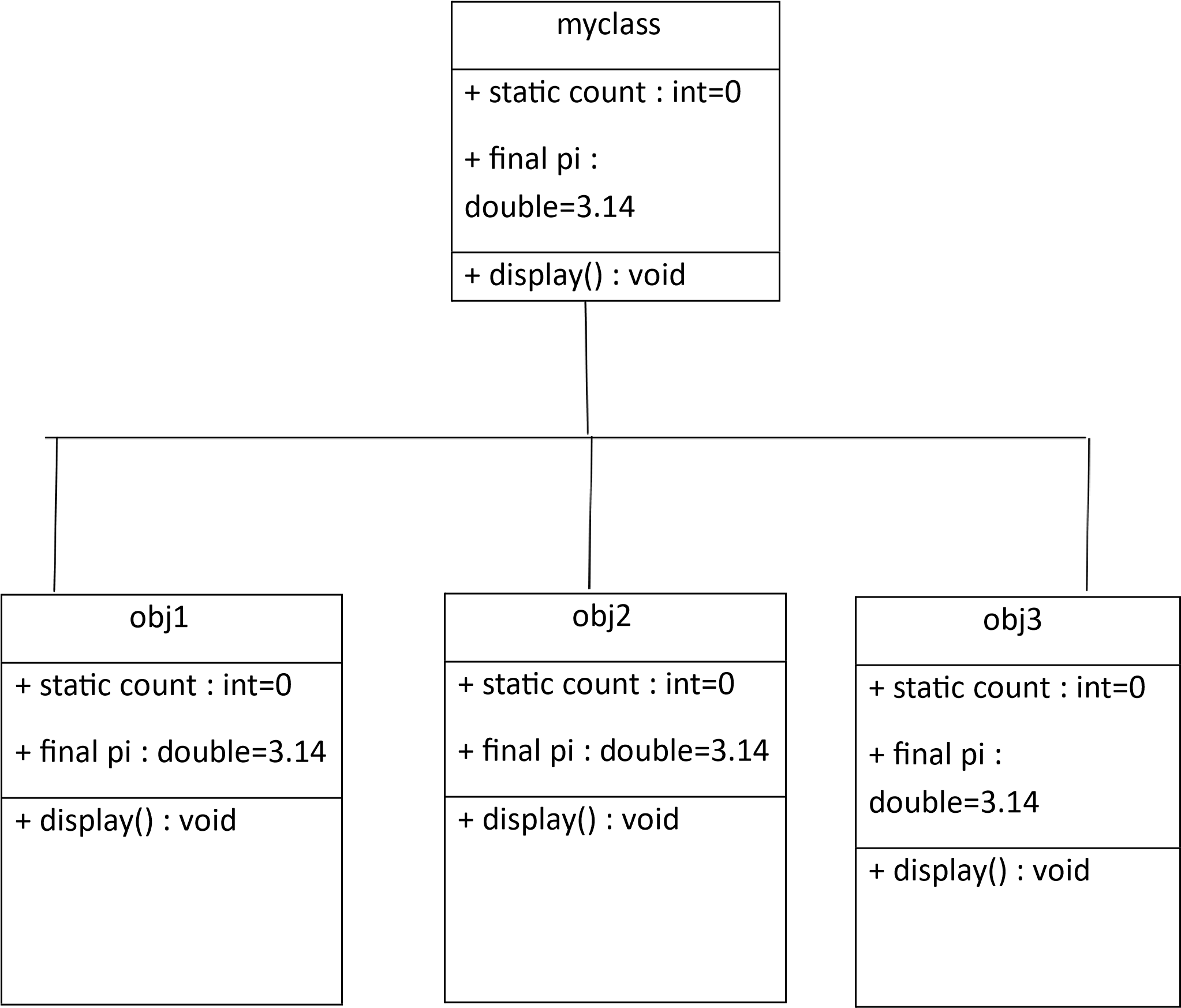
|  |  |  |
| --- | --- | --- |
| **S.No.** | **Expected Error** | **Reason** |
| **1** | **.variable** | **We must mention variable name to call the variable** |
| **2** | **static** | **Static variables contain only one value** |

**Important points:**

1.Declared a ‘static’ variable ‘count’ to keep the track of the number of objects are created static modifier indicates that the variable is a class level variable.

2.Declared a ‘final’ variable to ‘PI’ to represent a constant value ‘final’ modifier indicates that the variables value cannot be changed after it is initialized.

**3.The ‘count’ variable is ‘static’ ,so it can be accessed using the class name.**



2)Aim: Write a java program with class named “Book”. The class should contain various attributes such as “Title of the book , author , year of publication “. It should also contain a constructor with parameters details of the book.

i.e. “ Title of the book, author and year of publication”. Display the details of two books by creating two objects.

// Class representing a book

class Book {

// Attributes of the book

String title;

String author;

// Year of publication

int yop;

// Constructor to initialize book details

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

this.title = title;

this.author = author;

this.yop = yop;

}

// Method to display book details

void display() {

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

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

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

}

}

// Main class to test the Book class

class Details {

public static void main(String[] args) {

// Creating book objects with details

Book b1 = new Book("Staed", "Amul", 1910);

Book b2 = new Book("Python", "Madam", 2024);

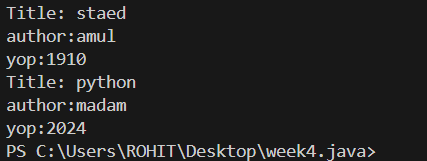
// Displaying book details

b1.display();

b2.display();

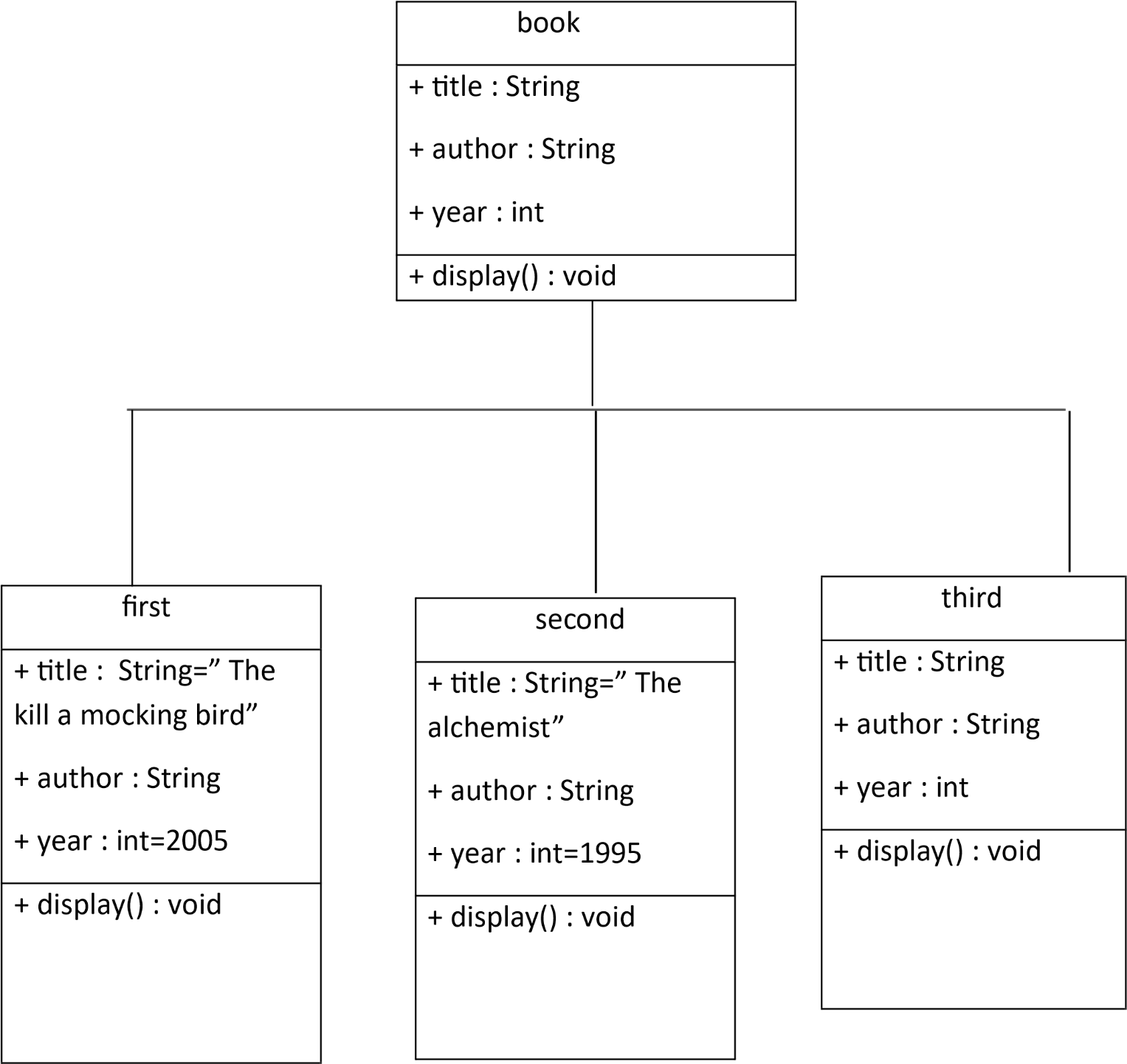
}

}

Output:

**Errors :**

|  |  |  |
| --- | --- | --- |
| S.No. | Expected Error | Reason |
| 1 | Setting the parameters inside the constructor | We cannot pass the values inside constructor without setting them first |

Class diagram:

Important points:

1.Constructors are special methods used to initialize objects.They have the same name as the class and no return type.It can have parameters to initialize the object with specific values.Here book is a constructor.

2.Methods are used to encapsulate functionality within a class.

**WEEK-5**1) **Create a calculator using the operations including addition, subtraction, multiplication and division using multilevel in heritance and display the desired output.**

- Write your code in VS CODE and execute

**- Important Points:**

* + - * Understand the calling of a Constructor
      * Giving class name correctly
      * Give the parameters Correctly

**CODE:**

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);

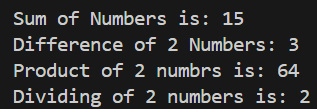
d.subtract(9,6);

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)** **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.**

**i. Cars should have an additional property: number of doors, Seating capacity.**

**ii. 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?**

**a. Truck should include and additional property capacity (in tons).**

**b. Create a showTruck() method to display the truck’s capacity.**

**c. 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.**- Write your code in VS CODE and execute

**- Important Points:**

* + - * Understand the calling of a Constructor
      * Giving class name correctly
      * Give the parameters Correctly

CODE: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("HONDA",120,5,5);

c.cardetails();

c.Details();

Bikes b=new Bikes("Ninja H2R",80,true);

b.bikedetails();

b.Details();

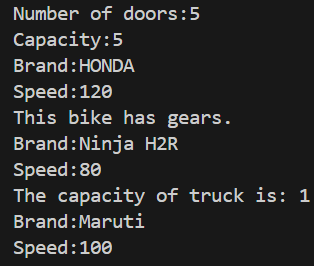
Trucks t=new Trucks("Maruti",100,1);

t.truckdetails();

t.Details();

}

}

**OUTPUT:**

**Errors:**

|  |  |  |
| --- | --- | --- |
| S.NO | Name | 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 |

**WEEK-6**

1) **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, model, fuel type, and color using the constructor**

- Write your code in VS CODE and execute

**- Important Points:**

* + - * Understand the calling of a Constructor
      * Giving class name correctly
      * Give the parameters Correctly

**CODE:**class Vehicle {

String Brand;

String model;

String fuel;

String color;

int capacity;

Vehicle(String Brand, String model, String fuel, int capacity, String color) {

this.Brand = Brand;

this.model = model;

this.fuel = fuel;

this.capacity = capacity;

this.color = color;

}

void displayInfo(String Brand, String model, String fuel, int capacity, String color) {

System.out.println("Vehicle Details: ");

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

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

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

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

System.out.println("Color: " + color);

}

}

class Car extends Vehicle {

Car(String Brand, String model, String fuel, int capacity, String color) {

super(Brand, model, fuel, capacity, color);

}

void displayInfo() {

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

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

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

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

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

System.out.println("Color: " + color);

}

}

class Week6\_1 {

public static void main(String[] args) {

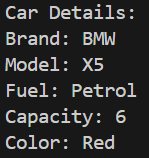
// Creating an instance of Car

Car car1 = new Car("BMW", "X5", "Petrol", 6, "Red");

car1.displayInfo(); // Display car details

}

}

**OUTPUT:**

**Errors:**

|  |  |  |
| --- | --- | --- |
| S.NO | **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) Create a Java program for the scenario.**

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

**i) UG admissions require a minimum of 60%**

**ii) PG admissions require a minimum of 70%**

- Write your code in VS CODE and execute

**- Important Points:**

* + - * Understand the calling of a Constructor
      * Giving class name correctly
      * Give the parameters Correctly

**CODE:**

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 week6\_2{

public static void main(String[] args){

UG ug=new UG();

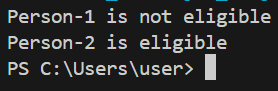
ug.geteligibility("Person-1",40);

PG pg=new PG();

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

}

}

OUTPUT:

**Errors:**

|  |  |  |
| --- | --- | --- |
| S.NO | 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 |

**3) Write a Java Program to create a Calculator class with overloaded methods to perform addition: Take the integer values a and b from the user.**

**i) Add two integers**

**ii) Add two doubles**

**iii) Add three integers**

- Write your code in VS CODE and execute

**- Important Points:**

* + - * Understand the calling of a Constructor
      * Giving class name correctly
      * Give the parameters Correctly

**CODE:**

class Calc{

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 week6\_3{

public static void main(String[] args){

Calc C1=new Calc();

System.out.println("Sum of 6 and 9 is: "+C1.add(6,9));

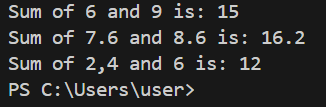
System.out.println("Sum of 7.6 and 8.6 is: "+C1.add(7.6,8.6));

System.out.println("Sum of 2,4 and 6 is: "+C1.add(2,4,6));

}

}

**OUTPUT:**



**Errors:**

|  |  |  |
| --- | --- | --- |
| S.NO | 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 |

4) Write a Java Program to 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.

- Write your code in VS CODE and execute

**-** Important Points:

* + - * Understand the calling of a Constructor
      * Giving class name correctly
      * Give the parameters Correctly

**CODE:**class Shape {

double calculateArea(double side) {

return side \* side;

}

double calculateArea(double width, double height) {

return width \* height;

}

}

class Circle extends Shape {

double calculateArea(double radius) {

return 3.14 \* radius \* radius;

}

}

class Week6\_4 {

public static void main(String[] args) {

Shape S1 = new Shape();

System.out.println("Area of square: " + S1.calculateArea(5));

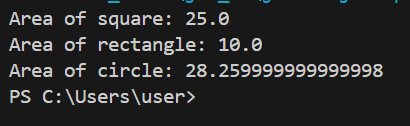
System.out.println("Area of rectangle: " + S1.calculateArea(2, 5));

Circle C1 = new Circle();

System.out.println("Area of circle: " + C1.calculateArea(3));

}

}

**OUTPUT:**

**Errors:**

|  |  |  |
| --- | --- | --- |
| S.NO | 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 |

WEEK:-7

**AIM:1:**Write a java program to create an abstract class animal with an abstract method called sound create subclass lion and tiger that extend the animal class and implement the sound method to make a specific sound for each animal

Code:

abstract  class Animal {

    abstract void Sound();

}

class lion extends Animal {

    public void Sound(){

        System.out.println("lRoar");

    }}

class Tiger extends Animal {

    public void Sound(){

        System.out.println("tRoar");

          }}

class main {

    public static void main(String[] args){

        System.out.println("NAME: HARI SIV SAI ROHIT.THOTA ,ROLL NO :AV.SC.U4CSE24126,SEC:CSE-B");

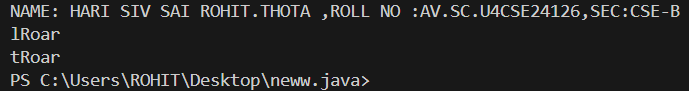
lion l =new lion();

l.Sound();

Tiger t = new Tiger();

t.Sound();

    }}

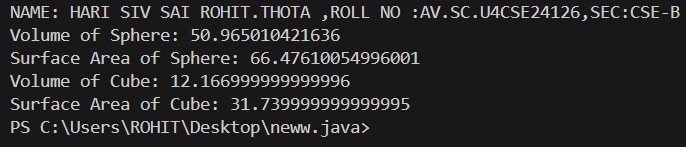
**OUTPUT:**

AIM:2: Write a java program to create an abstract class shape3D with abstract method calculatevolume () and calculatesurfacearea() create subclass sphere and cube that extends the shape3D class and implement the respective methods to calculate the volume and surface area of each shape.

Code:

|  |
| --- |
| abstract class Shape3D {          abstract double calculateVolume();          abstract double calculateSurfaceArea();      }        class Sphere extends Shape3D {          private double radius;            Sphere(double radius) {              this.radius = radius;          }            @Override          double calculateVolume() {              return (4.0 / 3.0) \* Math.PI \* radius \* radius \* radius;          }            @Override          double calculateSurfaceArea() {              return 4 \* Math.PI \* radius \* radius;          }      }        class  Cube extends Shape3D {          private double side;            Cube(double side) {              this.side = side;          }            @Override          double calculateVolume() {              return side \* side \* side;          }            @Override          double calculateSurfaceArea() {              return 6 \* side \* side;          }      }        class main {          public static void main(String[] args) {              System.out.println("NAME: HARI SIV SAI ROHIT.THOTA ,ROLL NO :AV.SC.U4CSE24126,SEC:CSE-B");              Sphere s = new Sphere(2.3);              System.out.println("Volume of Sphere: " + s.calculateVolume());              System.out.println("Surface Area of Sphere: " + s.calculateSurfaceArea());                Cube c = new Cube(2.3);              System.out.println("Volume of Cube: " + c.calculateVolume());              System.out.println("Surface Area of Cube: " + c.calculateSurfaceArea());          }      } |

OUTPUT:



AIM:3:write a java program using an abstract class to define a method for pattern printing.

Create an abstract class named pattern printer with an abstract method print pattern (int n) and a concrete method to display the pattern title

Implement two subclass:

* + - * 1. Star pattern: Print a right-angle triangle of stars(\*) .
        2. Number pattern : Print a right-angled triangle of increasing numbers.

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

Code:

abstract class PatternPrinter {

  abstract void printPattern(int n);

  void displayTitle(String title) {

      System.out.println("" + title);

      System.out.println("");

  }

}

class StarPattern extends PatternPrinter {

  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 {

  void printPattern(int n) {

      int number = 1;

      for (int i = 1; i <= n; i++) {

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

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

              number++;

          }

          System.out.println();

      }

  }

}

class Main {

  public static void main(String[] args) {

    System.out.println("NAME:HARI SIVA SAI ROHIT.THOTA , ROLL NO:24126,SEC:CSE-B ");

      int rows = 5;

      StarPattern star = new StarPattern();

      star.displayTitle("Star Pattern");

      star.printPattern(rows);

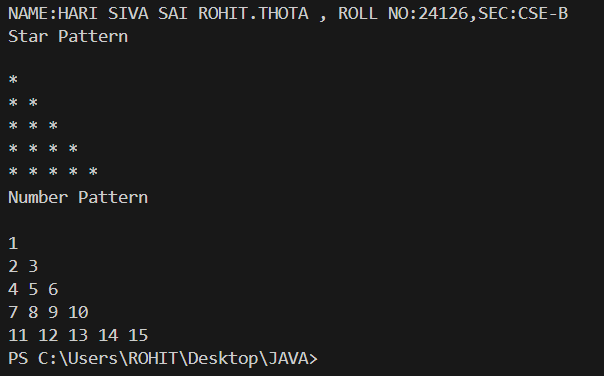
      NumberPattern number = new NumberPattern();

      number.displayTitle("Number Pattern");

      number.printPattern(rows);

  }}

OUTPUT:



WEEK-8

Write a java program to implement a login system using interfaces.

Code:

interface LoginSystem{

    boolean login(String id,String password);

}

class StudentPortal implements LoginSystem{

    public boolean login(String id, String password){

        if(id.equals("DIAMOND")&&password.equals("@3028")){

            System.out.println("Login Successful");

            return true;

        }

        else{

            System.out.println("Invalid Credentials");

            return false;

        }}}

class Main{

    public static void main(String[] args){

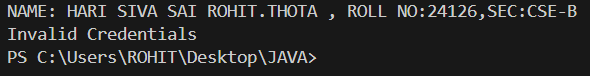
        System.out.println("NAME: HARI SIVA SAI ROHIT.THOTA , ROLL NO:24126,SEC:CSE-B ");

        StudentPortal portal = new StudentPortal();

        portal.login("Tony","@3028");

  }}

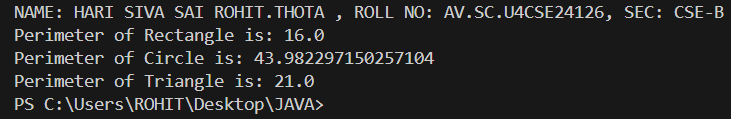
OUTPUT:



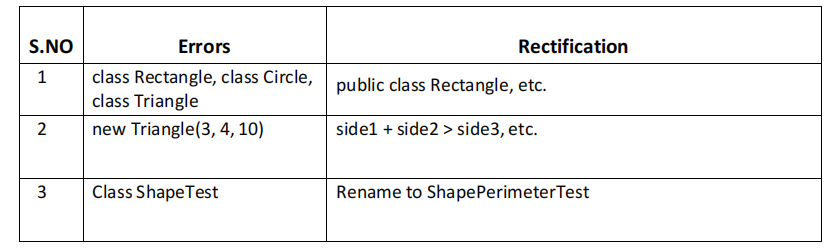
AIM:2:Write a java program to create an interface shape with the getperimeter() method.Create 3classes rectangle,circle and triangle that implements shape interface.Implement getperimeter() method for each of 3classes.

Code:

|  |
| --- |
| interface Shape {  double getPerimeter();  }  class Rectangle implements Shape {  private double length;  private double breadth;  Rectangle(double length, double breadth) {  this.length = length;  this.breadth = breadth;  }  public double getPerimeter() {  return 2 \* (length + breadth);  }}  class Circle implements Shape {  private double radius;  Circle(double radius) {  this.radius = radius;  }  public double getPerimeter() {  return 2 \* Math.PI \* radius;  }}  class Triangle implements Shape {  private double a, b, c;  Triangle(double a, double b, double c) {  this.a = a;  this.b = b;  this.c = c;  }  public double getPerimeter() {  return a + b + c;  }}  public class Main {  public static void main(String[] args) {  System.out.println("NAME: HARI SIVA SAI ROHIT.THOTA , ROLL NO: AV.SC.U4CSE24126, SEC: CSE-B");  Rectangle r = new Rectangle(5, 3);  System.out.println("Perimeter of Rectangle is: " + r.getPerimeter());  Circle c = new Circle(7);  System.out.println("Perimeter of Circle is: " + c.getPerimeter());  Triangle t = new Triangle(9, 8, 4);  System.out.println("Perimeter of Triangle is: " + t.getPerimeter());  }} |

**OUTPUT:**

**ERRORS:-**

****

IMPORTANT  POINTS:-

1. Interface Shape:
2. The interface Shape contains one method getPerimeter(), which will be implemented by any class that implements this interface.
3. Rectangle Class:

a.The Rectangle class implements the Shape interface.

b.The perimeter of a rectangle is calculated using the formula:   
Perimeter = 2 \* (length + width)

c.The getPerimeter() method returns the perimeter of the rectangle.

1. Circle Class:
2. The Circle class implements the Shape interface.

b.The perimeter (circumference) of a circle is calculated using the formula:   
Circumference = 2 \* π \* radius

 c.The getPerimeter() method returns the circumference of the circle.

1. Triangle Class:
2. The Triangle class implements the Shape interface.

b.The perimeter of a triangle is the sum of its three sides:   
Perimeter = side1 + side2 + side3

c.The getPerimeter() method returns the perimeter of the triangle.

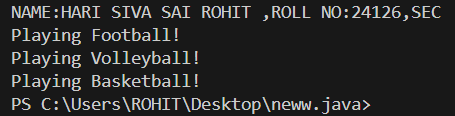
1. Main Class:In the main method, we create objects for Rectangle, Circle, and Triangle and call their respective getPerimeter() methods to display the perimeters.

AIM:3:Write a java program to create an interface label with method play that takes no arguments and return void create 3 classes football,volley ball and basketball that implements interface and override play() method to play respective sports.

Code:

|  |
| --- |
| interface Playable {      void play();  }  class Football implements Playable {      @Override      public void play() {          System.out.println("Playing Football!");      }}  class Volleyball implements Playable {      @Override      public void play() {          System.out.println("Playing Volleyball!");      }  }  class Basketball implements Playable {      @Override      public void play() {          System.out.println("Playing Basketball!");      }}  class main {      public static void main(String[] args) {          System.out.println("NAME:HARI SIVA SAI ROHIT ,ROLL NO:24126,SEC:CSE-B");          Playable football = new Football();          Playable volleyball = new Volleyball();          Playable basketball = new Basketball();          football.play();          volleyball.play();          basketball.play();      }} |

**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Error** | **Rectification** |
| **1.** | **Football is not abstract and does not override abstract method play()** | **Implement the play method from interface** |
| **2.** | **Playables.java but public class play test** | **Class play test is public,should be declared in a file named playtest.java** |

**IMPORTANT POINTS:-**

* The Playable interface defines the contract (play() method).
* Each class (Football, Volleyball, Basketball) implements the interface and provides a specific behavior for play().
* This demonstrates polymorphism using interfaces in Java.