**23CSE111**

**LAB MANUAL**



**Department of CSE**

**Amrita School of Engineering**

**Amrita Vishwa Vidyapeetham, Amaravati Campus**

**Verified By :- Name: M.ARCHITH**

**Roll No: AV.SC.U4CSE24223**

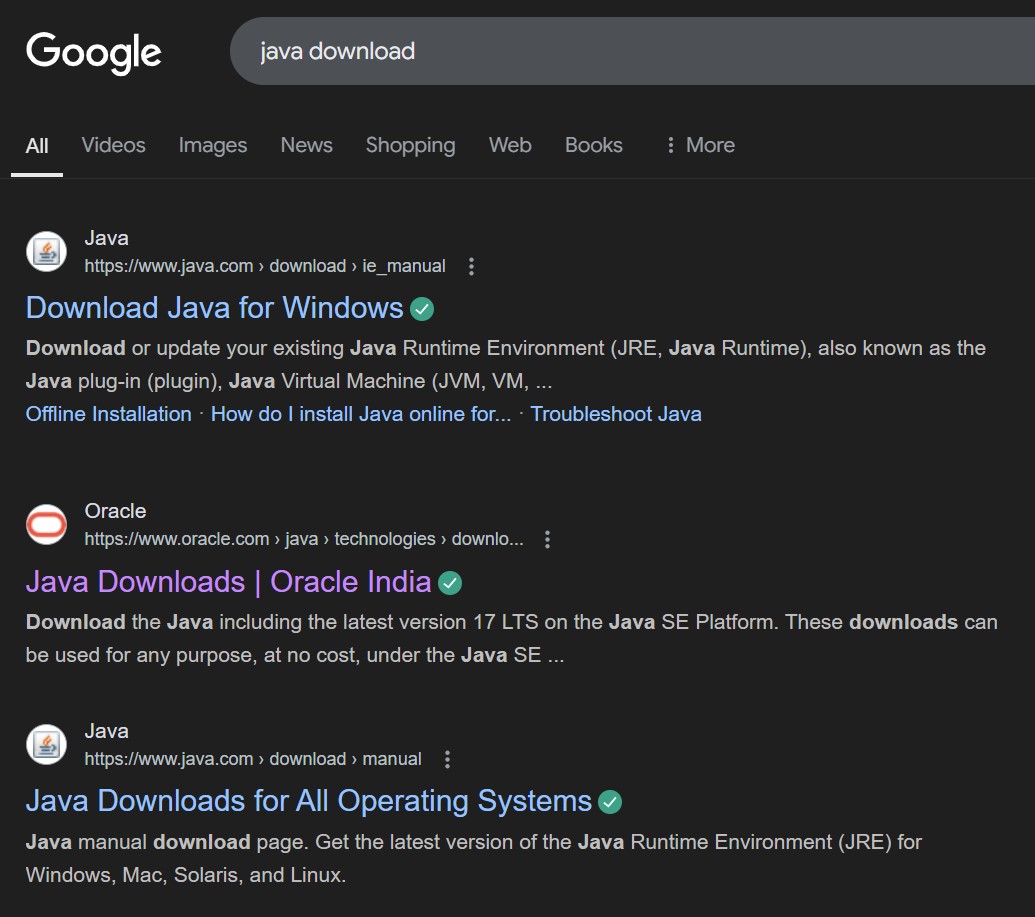
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.NO | Programs | Date | Pg:No | Signature |
| 1 | c)Download and Install Java Software.  b)Write a java program to print message “Welcome to java programming”.  c)Write a java program that prints name,roll number,section of a student. |  |  |  |
| 2 | a) To write a java program on simple interest and by taking input from user  b) To write a java program on factorial of a number by taking input from user  c) To write a java program to convert temperature from Fahrenheit to Celsius  d) To write a java program to calculate the fibonacii sequence and take the input from user  e) To write a java program to find the area of triangle by using heron’s formula taking the input from the user  f) write java program to find the area of rectangle |  |  |  |
| 3 | a)To create a java program with following instructions  1.create a class with name car  2.create a four attributes named car\_brand,car\_color,fuel\_type,mileage.  3. Create three methods named start(), stop(). Service()  4. Create three objects named car1,car2 and car3  b) To create a class bank account with methods deposit() and withdrawl  c) |  |  |  |
| 4 | a)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.  b) write a java program with class named “myclass” with a static variable count of int type. intialize it to zero and a constant variable “pi” of type double initialized to “3.14”  as attributes of that class. now define a constructor for “myclass”, that increments the count variable each time an object of “myclass” is created. finally, print the final values  of ‘count’ and ‘pi’ variables and create 3 objects |  |  |  |
| 5 | a) 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 Constructor 2. Giving class name correctly 3. Give the parameters Correctly  b) 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 |  |  |  |
| 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 .   b)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%  c) 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  d) 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.  - Important Points:  1.Understand the calling of a Constructor  2.Giving class name correctly  3.Give the parameters Correctly |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Week-1

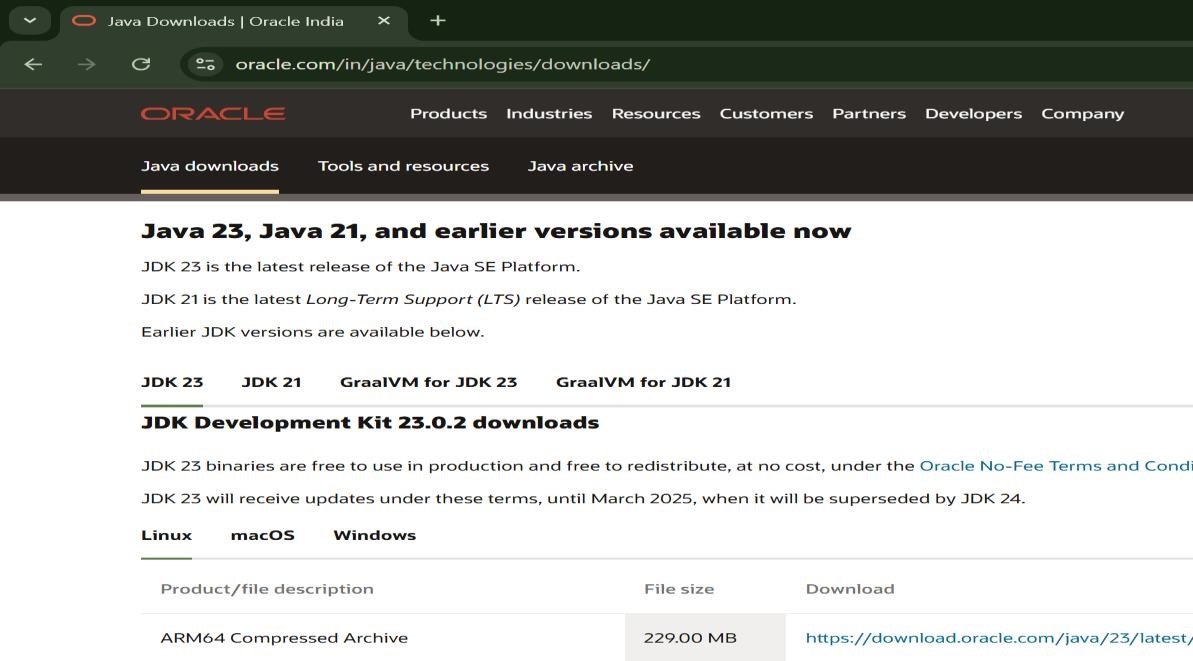
# Program-1:-

## Aim:- to Download and Instal the Java Software and

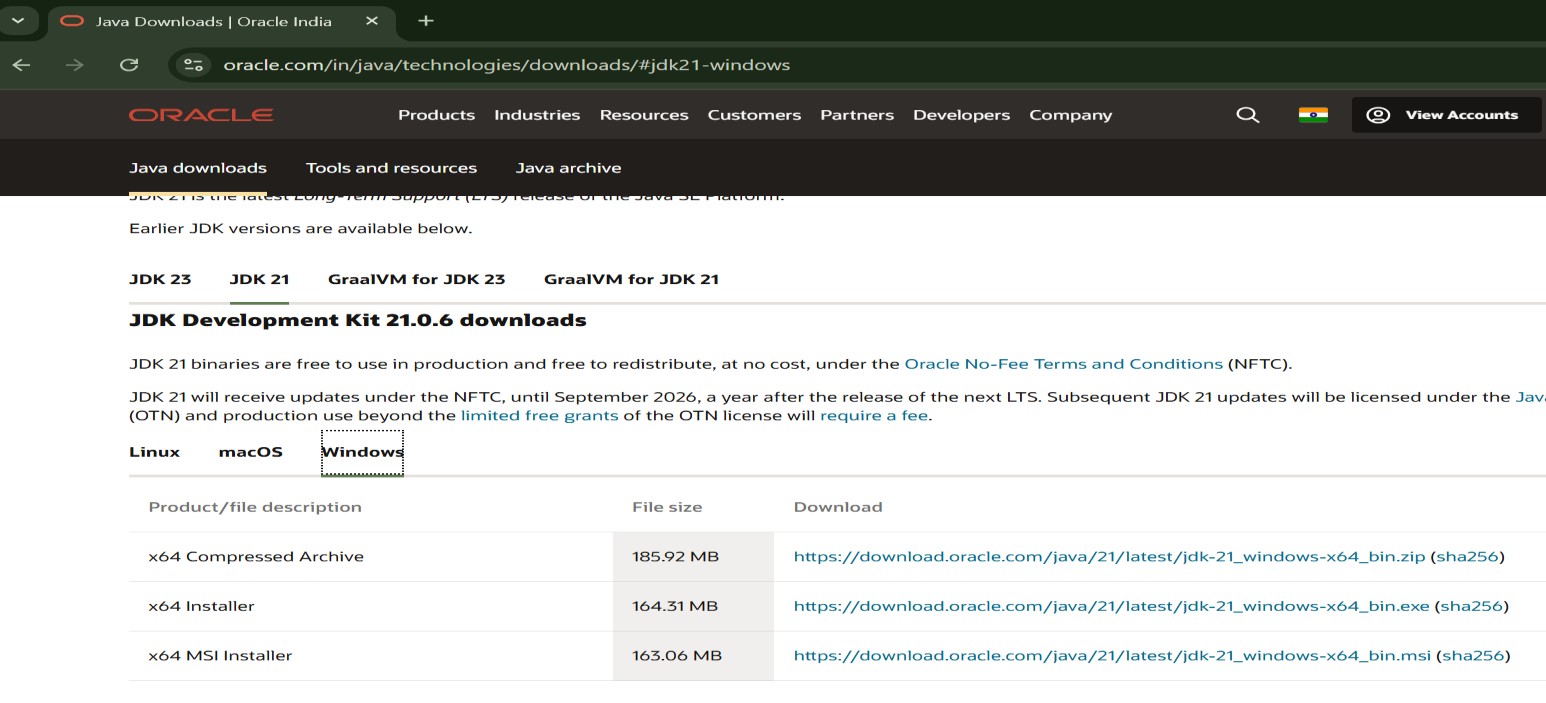
## Procedure: Step-1:- Type Java download in search



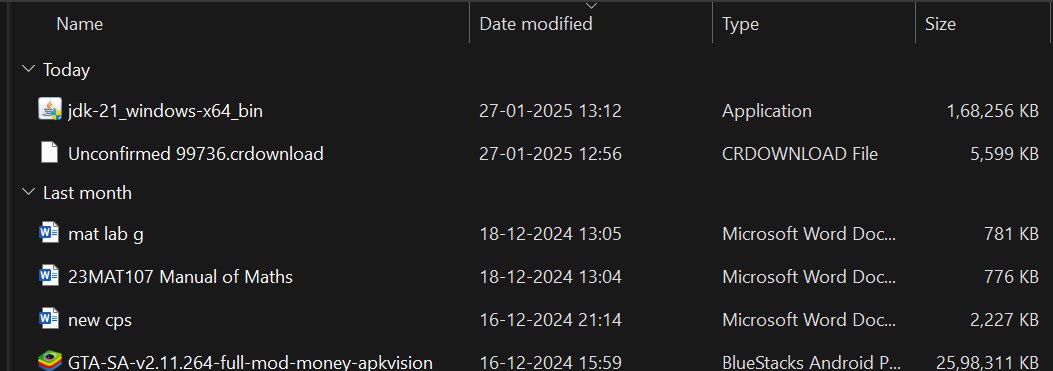
## Step-2:-click on oracle java download and enter into oracle website



## Step-3:-click on JDK21 and click on windows and later click on x64 instalier link to download

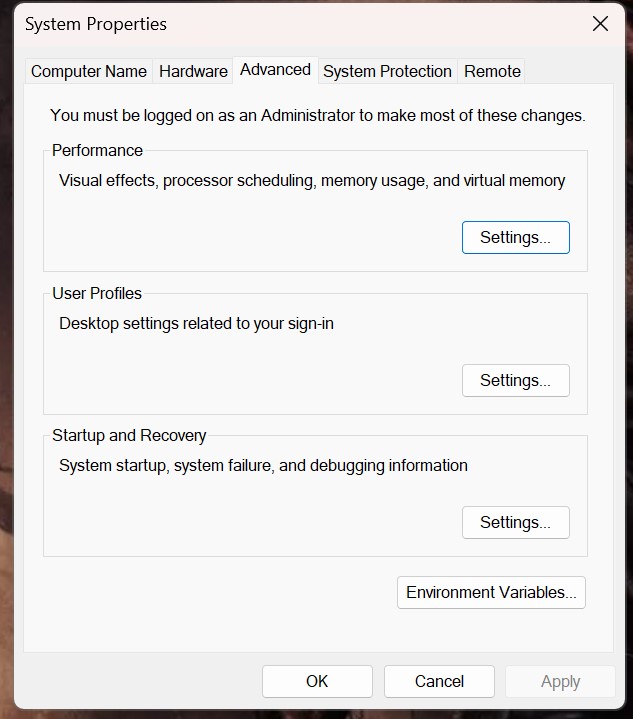


## Step-4:-After completing download click on it’s file and then give permission to install



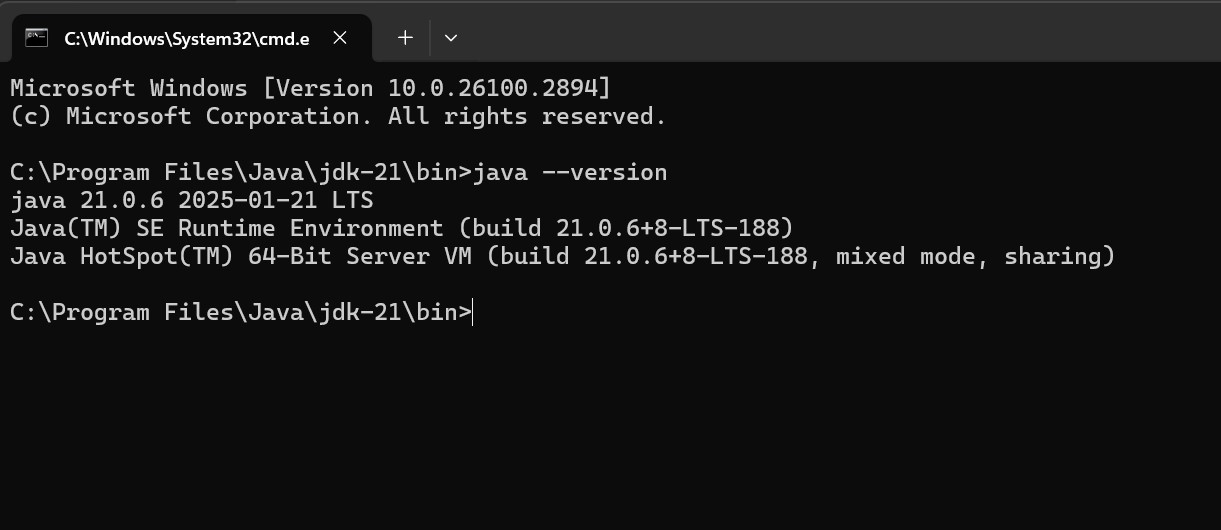
Step-5:-Then go to (This pc) in that click (windows{c}) in that click (Program files) in that click (Java) in that click (jdk-21) in that click (bin)

## Step-6:-Select and copy path of opening the file and then press windows and search System Environmental



Step-7:-After opening Environment variables then past path of opening file in user variable and click on ok

## Step-8:-To verify version open CMD and type java --version



**Program : 2**

## Aim:-write a java program to print[welcome to java programming] Input:-

class ex\_1{

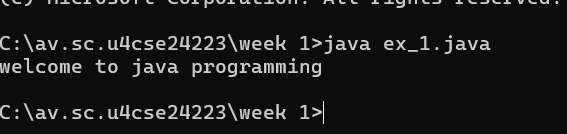
public static void main(String[] args){

System.out.println("welcome to java programming");

}

}

## Output:-



### Program : 3

Aim:-write a java program that prints name, roll no, section of the student Input:-

class Studentdet{

public static void main(String[] args){

System.out.println("Name= archith”);

System.out.println("section= cse-c");

System.out.println("rollno= 24223");

}

}

### Output:-



Week-2

Program:1

Aim: To write a java program on simple interest and by taking input from user

import java.util.Scanner;

public class SI {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the principal amount: ");

        Float principal = scanner.nextFloat();

        System.out.println("Enter the interest rate: ");

        Float rate = scanner.nextFloat();

        System.out.println("Enter the time period: ");

        Float time = scanner.nextFloat();

        Float interest = (principal \* rate \* time) / 100;

        Float totalAmount = principal + interest;

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

        System.out.println("Total Amount: " + totalAmount);

    }

}

Output:

A screenshot of a computer program

AI-generated content may be incorrect.

Error table:

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Error | Reason of error | rectification |
| 1 | Syntax error | ; missing | ; added |
| 2 | Logical error | Wrong formula (principal\*rate\*time) | Corrected formula  by (principal\*rate\*time)/100 |

Program:2

Aim: To write a java program on factorial of a number by taking input from user

import java.util.Scanner;

public class Factorial {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number to calculate its factorial: ");

int number = scanner.nextInt()

if (number < 0) {

System.out.println("Factorial is not defined for negative numbers.");

} else {

long factorial

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

factorial \*= i;

}

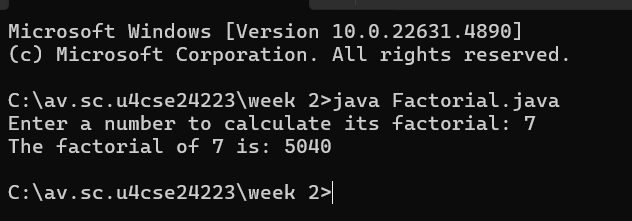
System.out.println("The factorial of " + number + " is: " + factorial);

}

scanner.close();

}

}

Output: 

Error table:

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Error | Reason of error | Rectification |
| 1 | Syntax error | } error | Fixed with keeping } |
| 2 | Missing import statement | Didn’t importing packages | Packages imported |
| 3 | Run time | Incorrect path | Fixed with correct path |

Program:3

Aim: To write a java program to convert temperature from Fahrenheit to Celsius

import java.util.Scanner;

public class Temp {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

float fahrenheit = scanner.nextFloat();

float celsius;

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

System.out.println("The temperature in Celsius is: " + celsius);

scanner.close();

}

}

Output:

A screenshot of a computer program

AI-generated content may be incorrect.

Error table:

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Error | Reason of error | Rectification |
| 1 | Logical error | Incorrect formula celsius = (fahren - 32) \* 5/9; | Corrected by formula Celsius=(fahren-32)\*5/9 |
| 2 | Missing import statement | Util package is missing | Util package is added |

Program:4

Aim: To write a java program to calculate the fibonacii sequence and take the input from

User

import java.util.Scanner;

public class Fib {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

if (n <= 0) {

System.out.println("Please enter a positive integer.");

} else {

long f1 = 0, f2 = 1;

System.out.print(f1 + " ");

if (n > 1) {

System.out.print(f2 + " ");

}

for (int i = 3; i <= n; i++) {

long f3 = f1 + f2;

System.out.print(f3 + " ");

f1 = f2;

f2 = f3;

}

}

scanner.close();

}

}

Output: A screen shot of a computer

AI-generated content may be incorrect.

Error table:

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Error | Reason of error | Rectification |
| 1 | Logical error | Incorrect formula | Fixed with correct formula |
| 2 | Run time | Incorrect path | Added correct path |

Program:5

Aim: To write a java program to find the area of triangle by using heron’s formula taking

the input from the user

import java.util.Scanner;

public class Heron {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

double a = scanner.nextDouble();

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

double b = scanner.nextDouble();

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

double c = scanner.nextDouble();

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

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

if (a + b > c && b + c > a && c + a > b) {

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

} else {

System.out.println("it do not from triangle.");

}

scanner.close();

}

}

Output: A computer screen shot of a black screen

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

Error table:

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Error | Reason for error | Rectification |
| 1 | Logical error | Incorrect formula | Fixed with correct formula |
| 2 | Name error | Undeclared variable | Variable declared |

Program:6

Aim: write java program to find the area of rectangle:

import java.util.Scanner;

public class Rectangle {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in)

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

double length = scanner.nextDouble();

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

double width = scanner.nextDouble();

double area = length \* width;

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

scanner.close();

}

}

Output: A screen shot of a computer

AI-generated content may be incorrect.

Error table:

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | error | Reason of error | Rectification |
| 1 | Syntax | ; missing | ; added |
| 2 | Logical error | Incorrect formula | Fixed with correct formula |

Week-3

Program:1

Aim: To create a java program with following instructions

1.create a class with name car

2.create a four attributes named car\_brand,car\_color,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[]){

car car1 = new car();

car1.Car\_color = "white";

car1.Car\_brand = "tata";

car1.fuel\_type = "ev";

car1.mileage = 70;

car1.start();

car car2 = new car();

car2.Car\_color = "red";

car2.Car\_brand = "ferrari";

car2.fuel\_type = "petrol";

car2.mileage = 200;

car2.stop();

car car3 = new car();

car3.Car\_color = "black";

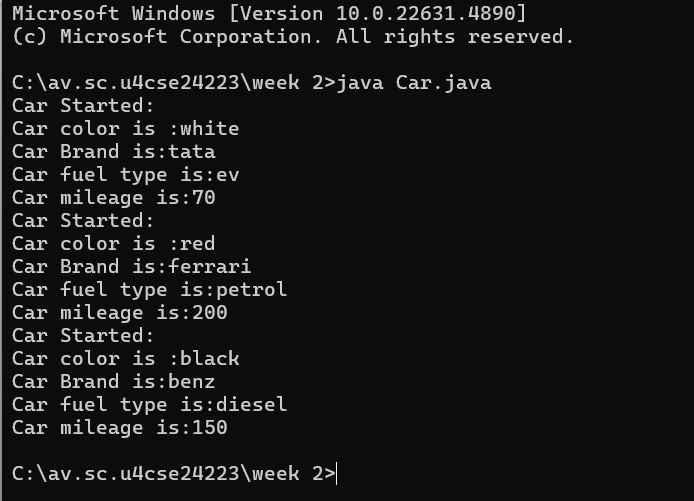
car3.Car\_brand = "benz";

car3.fuel\_type = "diesel";

car3.mileage = 150;

car3.service();

}

Output: 

Error table:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | error | Reason of error | rectification |
| 1 | Syntax error | Type in the variable name mileage in the constructor | Corrected to mileage |

Class diagram:

|  |
| --- |
| Car |
| +carcolor: string  -carbrand:string  +mileage:int |
| +car():void  +start():void  +stop():void  +service:void |

Program:2

Aim:To create a class bank account with methods deposit() and withdrawl

Class bank account{

public void withdraw(int amount) {

if (amount <= balance) {

balance -= amount;

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

+balance);

} else {

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

}

}

public void deposit(int amount) {

balance += amount;

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

balance);

}

public void displayDetails() {

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

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

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

}

public void blackbox(int withdrawAmount, int depositAmount) {

withdraw(withdrawAmount);

deposit(depositAmount);

displayDetails();

}

public static void main(String[] args) {

Bank\_Account a = new Bank\_Account("ARCHITH", 45678, 1324);

a.blackbox(3000, 200);

System.out.println();

Bank\_Account a1 = new Bank\_Account("ROHITH", 60000, 18000)a1.blackbox(7000,50000);

}

}

Output: A computer screen shot of a black screen

AI-generated content may be incorrect.

Errors:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | error | Reason of error | rectification |
| 1 | Missing balance dispaly | Withdraw and deposit methods update the | Add system.out.println on both methods |

Class diagram:

|  |
| --- |
| bank |
| -name:string  -accno:string  Currbal:string |
| +bankAccount():void  +deposit():int  +withdraw():void |

Week-4

Program: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.

class book{

    public String title;

    public String author;

    public int year\_of\_publication;

    public void start() {

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

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

        System.out.println("Year of publication of the book is :"+year\_of\_publication);

    }

    public void service() {

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

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

        System.out.println("Year of publication of the book is :"+year\_of\_publication);

    }

    public static void main(String[] args){

        book book1=new book();

        book1.title="quantum theory";

        book1.author="Louis de Broglie";

        book1.year\_of\_publication=1924;

        book1.start();

        book book2=new book();

        book2.title="A Brief History of Time";

        book2.author="Stephen hawkimg";

        book2.year\_of\_publication=1988;

        book2.service();

        }

 }

Output:

A computer screen with white text

AI-generated content may be incorrect.

Errors:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | error | Reason of error | rectification |
| 1 | Compilation error | Missing semicolon | Add semicolons at the end of the lines |
| 2 | Logical error | Missing spacing in the print statements | Add a space after year of publication in the display details () method |

Classdiagram:

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

Program:2

Aim: write a java program with class named “myclass” with a static variable count of int

type. intialize it to zero and a constant variable “pi” of type double initialized to “3.14”

as attributes of that class. now define a constructor for “myclass”, that increments the

count variable each time an object of “myclass” is created. finally, print the final values

of ‘count’ and ‘pi’ variables and create 3 objects

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();

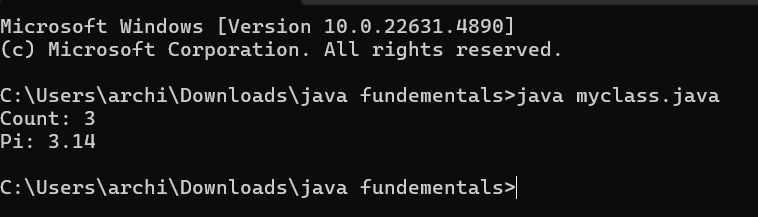
System.out.println("Count: "+count);

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

}

}

Output:



Errors:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | error | Reason of error | rectification |
| 1 | Logical error | Missing space in the println statement for count and pi | Add a space before +count and+pi in the println statement |

Class diagram:

|  |
| --- |
| MyClass |
| - count: int (static)  - pi: double (static final) |

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

Program1:

class caluclator{

public void add(int a,int b){

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

 }

 public void subtract(int a,int b){

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

 }

}

class multi extends caluclator{

public void product(int a,int b){

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

 }

}

class div extends multi{

 public 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){

 div d=new div();

 d.add(7,4);

 d.subtract(9,3);

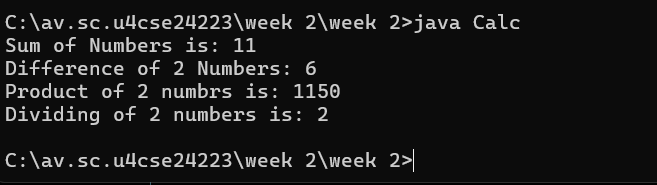
 d.product(23,50);

 d.divide(4,2);

}

 }

Output:



Errors:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | error | Reason of error | rectification |
| 1 | Compalation/sytax | Forgot to keep semicolon | Absence of Semicolon |
| 2 | Class Name Error | Forgot to keep  Capital letter in class | Give the class name correctly |

Program 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 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.

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("bugati",120,5,5);

c.cardetails();

c.Details();

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

b.bikedetails();

b.Details();

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

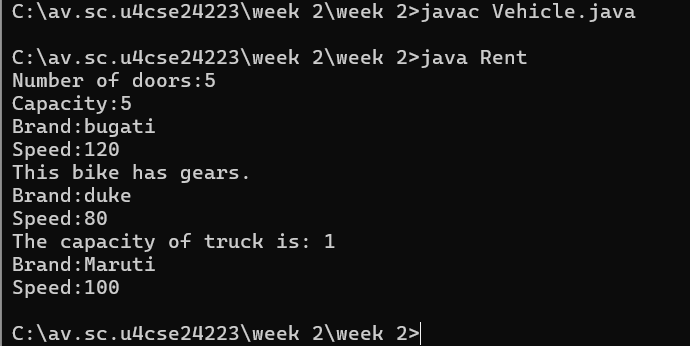
t.truckdetails();

t.Details();

}

}

Output:



Errors:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | error | Reason of error | rectification |
| 1 | Closing Brackets | Forgot to keep close brackets | Need to Close the brackets |
| 2 | Class Name Error | Forgot to keep  Capital letter in class | Give the class name correctly |

Week-6:

Program-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, model, fuel type, and

color using the

constructor .

Important Points:

1.Understand the calling of a Constructor

2.Giving class name correctly

3.Give the parameters Correctly

class vehicales {

String Brand;

String model;

String fuel;

String color;

int capacity;

vehicales(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 vehicales {

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 sub {

public static void main(String[] args) {

Car car1 = new Car("auid", "q5", "Petrol", 6, "black");

car1.displayInfo();

}

}

Output:

A black screen with white text

AI-generated content may be incorrect.

Errors:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | error | Reason of error | rectification |
| 1 | Closing Brackets | Forgot to keep close brackets | Need to Close the brackets |
| 2 | Class Name Error | Forgot to keep  Capital letter in class | Give the class name correctly |
| 3 | Constructor Calling | Called the constructor wrongly | Call the constructor correctly |

Class diagram:

|  |
| --- |
| Vehicle |
| +displayInfo() |



|  |
| --- |
| Car |
| - model: String  - fuelType: String  - color: String |
| +Car(model, fuelType, color)  +displayInfo() |

Program-2:

Aim: 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%

Important Points:

1.Understand the calling of a Constructor

2.Giving class name correctly

3.Give the parameters Correctly

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 scen{

public static void main(String[] args){

UG ug=new UG();

ug.geteligibility("Person-1",60);

PG pg=new PG();

pg.geteligibility("Person-2",70);

}

}

Output:

Errors:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | error | Reason of error | rectification |
| 1 | Closing Brackets | Forgot to keep close brackets | Need to Close the brackets |
| 2 | Class Name Error | Forgot to keep  Capital letter in class | Give the class name correctly |



Class diagram:



|  |
| --- |
| Student |
| - name: String  - percentage: double |
| +Student(name, percent)  +displayInfo()  +isEligible(): boolean |



|  |
| --- |
| UGAdmission |
| +isEligible(): boolean |

|  |
| --- |
| PGAdmission |
| +displayInfo() +displayInfo() |

Aim: 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

- Important Points:

1.Understand the calling of a Constructor

2.Giving class name correctly

3.Give the parameters Correctly

Program-3:

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 try{

public static void main(String[] args){

Calc C1=new Calc();

System.out.println("Sum of 7 and 5 is: "+C1.add(7,5));

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

System.out.println("Sum of 1,3 and 9 is: "+C1.add(1,3,9));

}

}

Output:

A black screen with white text

AI-generated content may be incorrect.

Errors:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | error | Reason of error | rectification |
| 1 | Closing Brackets | Forgot to keep close brackets | Need to Close the brackets |
| 2 | Class Name Error | Forgot to keep  Capital letter in class | Give the class name correctly |

Class diagram:

|  |
| --- |
| Calculator |
| + add(int, int): int  + add(double, double): double  + add(int, int, int): int |

Program-4:

Aim: 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.

- Important Points:

1.Understand the calling of a Constructor

2.Giving class name correctly

3.Give the parameters Correctly

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 c4 {

public static void main(String[] args) {

Shape S1 = new Shape();

System.out.println("Area of square: " + S1.calculateArea(7));

System.out.println("Area of rectangle: " + S1.calculateArea(2, 7));

Circle C1 = new Circle();

System.out.println("Area of circle: " + C1.calculateArea(4));

}

}

Output:

A screen shot of a computer

AI-generated content may be incorrect.

Errors:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | error | Reason of error | rectification |
| 1 | Closing Brackets | Forgot to keep close brackets | Need to Close the brackets |
| 2 | Class Name Error | Forgot to keep  Capital letter in class | Give the class name correctly |
| 3 | Compalation/sytax | Forgot to keep semicolon | Absence of Semicolon |

Class diagram:

|  |
| --- |
| Shape |
| + calculateArea(): double  + calculateArea(double side): double  +calculateArea(double l, double w): double |



|  |
| --- |
| Circle |
| - radius: double |
| + Circle(radius: double)  + calculateArea(): double |

Week-7

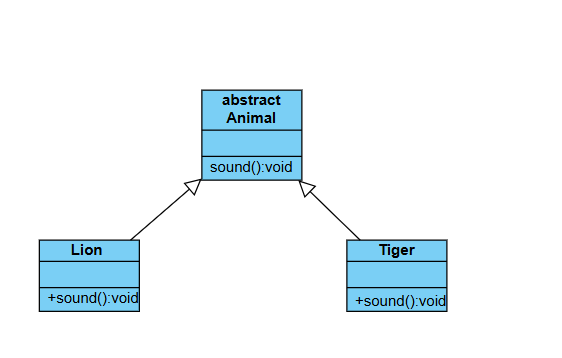
Program:1

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.

Class diagram:



Code:

abstract class Animal{

    public abstract void sound();

}

class Lion extends Animal{

    public void sound(){

        System.out.println("lion sounds like roar");

    }

}

class Tiger extends Animal{

    public void sound(){

        System.out.println("tiger sounds like growl");

    }

}

class Main{

    public static void main(String[] args){

        System.out.println("Name: archith rollno:av.sc.u4cse24223 class:cse-c");

        Tiger tiger=new Tiger();

        tiger.sound();

  Lion lion = new Lion();

        lion.sound();

    }

}

Output:

A black screen with white text

AI-generated content may be incorrect.

ERROR TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | Error | Reason of error | rectification |
| 1 | Case sensitivity | new tiger() uses lowercase t but class name is Tiger | Change new tiger() to new Tiger() |

IMPORTANT POINTS:

1. Animal class is declared as an abstract.

2.the method sound has no parameters in super class and subclasses.

3.each subclass extends from super class.

Program:2

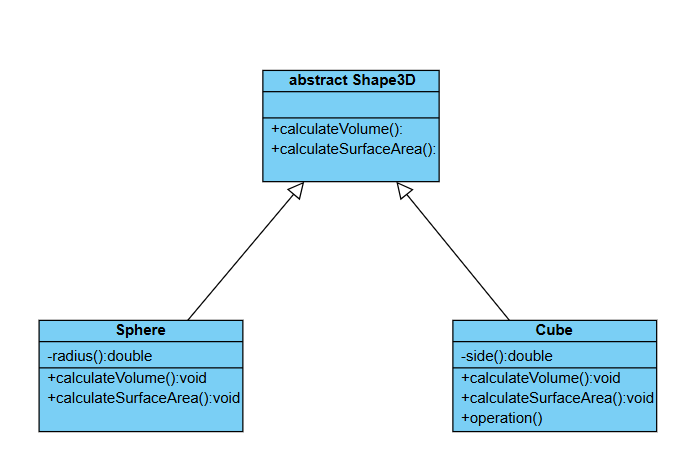
AIM: Write a Java program to create an abstract class Shape3D with abstract methods

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.

CLASS DIAGRAM:



CODE:

abstract class Shape3D{

    public abstract void calculateVolume();

    public abstract void calculateSurfaceArea();

}

class Sphere extends Shape3D{

    private double radius;

    public Sphere(double radius){

        this.radius=radius;

    }

    public void calculateVolume(){

        System.out.println("volume of sphere is :"+(4.0/3.0)\*3.14\*radius\*radius\*radius);

     }

     public void calculateSurfaceArea(){

        System.out.println("surface area of sphere:"+4\*3.14\*radius\*radius);

}

}

class Cube extends Shape3D{

    private double side;

    public Cube(double side){

        this.side=side;

    }

    public void calculateVolume(){

        System.out.println("volume of cube is:"+side\*side\*side);

    }

    public void calculateSurfaceArea(){

        System.out.println("surface area of cube is:"+6\*side\*side);

    }

}

class Main{

    public static void main(String[] args){

        System.out.println("Name:archith roll no:av.sc.u4cse24223");

        Sphere s=new Sphere(5.0);

        s.calculateVolume();

        s.calculateSurfaceArea();

        Cube c=new Cube(5.0);

        c.calculateVolume();

        c.calculateSurfaceArea();

    }

}

Output:

A screen shot of a computer code

AI-generated content may be incorrect.

ERROR TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | eorrr | Reason of error | rectification |
| 1 | Syntax Error | Missing semicolon after this.side = side in Cube constructor | Add ; after this.side = side; |

IMPORTANT POINTS:

1. Shape3D is an abstract class that defines a contract for 3D shapes.

2. It contains two abstract methods: calculateVolume() and calculateSurfaceArea()

3. Sphere and Cube classes extend Shape3D and override both methods.

Program:3

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

create an abstract class named PatternPrinter with an abstract method PrintPattern(int n) and

a concrete method to display the pattern title

Implement two subclasses:

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 patterns for a given

number of rows

class diagram:

A diagram of a pattern

AI-generated content may be incorrect.

CODE:

abstract class PatternPrinter{

    public abstract void printPattern(int n);

    public void display(String str){

        System.out.println("n " + str);

        System.out.println("----------------");

    }

 }

class StarPattern extends PatternPrinter{

    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{

    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();

        }

    }

}

class Main{

    public static void main(String[] args){

        System.out.println("Name:archith rollno:av.sc.u4cse24223 class:cse-c ");

        StarPattern s =new StarPattern();

        s.display("Star Pattern");

        s.printPattern(5);

        NumberPattern n = new  NumberPattern();

        n.display("number pattern");

        n.printPattern(5);

    }

}

Output:

A computer screen with white text

AI-generated content may be incorrect.

ERROR TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | error | Reason of error | rectification |
| 1 | |  | | --- | |  |   Syntax Error | Missing closing double quote in s.display("Star Pattern); | Change to s.display("Star Pattern"); |

IMPORTANT POINTS:

1. Abstract Class (PatternPrinter)
2. Abstract Method printPattern(int n) (must be implemented by subclasses).

Concrete Method  displayTitle(String title) (prints a formatted pattern title).

Cannot be instantiated directly (must be extended).

Week-8

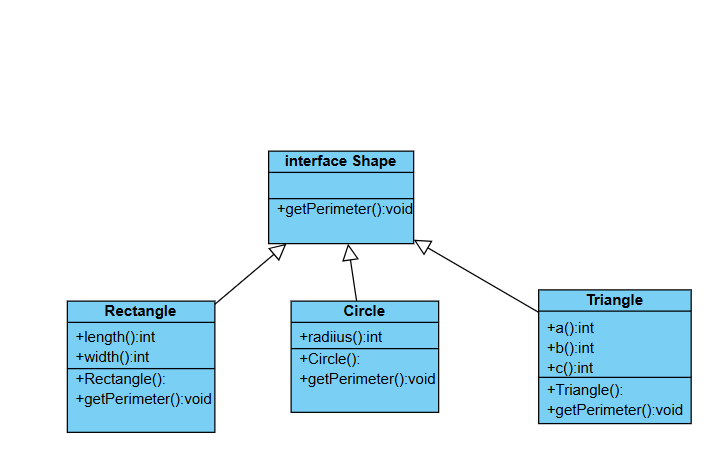
Program:1

AIM: Write a java program to create an interface Shape with the getPerimeter() method .

Create three classes Rectangle, Circle, Triangle that implement the Shape interface ,Implement

the getPerimeter() method for each of the three classes.

CLASS DIAGRAM:



Code:

interface Shape{

    void getPerimeter();

}

class Rectangle implements Shape{

    int length;

    int width;

    public Rectangle(int length,int width){

        this.length=length;

        this.width=width;

    }

    public void getPerimeter(){

        System.out.println("perimeter of rectangle:"+2\*(length+width));

    }

}

class Circle implements Shape{

    int radius;

    public Circle(int radius){

        this.radius=radius;

    }

    public void getPerimeter(){

        System.out.println("perimeter of circle:"+2\*3.14\*radius);

    }

}

class Triangle implements Shape{

    int a;

    int b;

    int c;

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

        this.a=a;

        this.b=b;

        this.c=c;

    }

    public void getPerimeter(){

        System.out.println("perimeter of triangle:"+a+b+c);

    }

}

class Main{

    public static void main(String[] args){

        System.out.println("Name:archith rollno:av.sc.u4cse24223 class:cse-c ");

        Triangle t=new Triangle(4,2,5);

        t.getPerimeter();

        Circle c=new Circle(6);

        c.getPerimeter();

        Rectangle r=new Rectangle(7,8);

        r.getPerimeter();

    }

}

Output:

A screen shot of a black background

AI-generated content may be incorrect.

ERROR TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | Error type | Reason of error | rectification |
| 1 | Syntax Error | circle is written in lowercase when creating object: circle c = new Circle(5); | Correct to Circle c = new Circle(5); (capital C) |

IMPORTANT POINTS:

1. Shape is an interface with the method void getPerimeter();
2. All shapes (Rectangle, Circle, Triangle) implement the Shape interface.
3. Each class overrides getPerimeter() method from the Shape interface.

4.Each shape class has its own constructor to initialize dimensions.

Program:2

AIM: Write a java program to create an interface playable with a method play() that takes no

arguments and return void. Create three classes Football, Volleyball, and Basketball that

implement the playable interface and override the play() method to play the respective sports

class diagram:

A diagram of a volleyball player

AI-generated content may be incorrect.

CODE:

interface Playable{

    void play();

}

class Football implements Playable{

    public void play(){

        System.out.println("playing a football");

    }

}

class Volleyball implements Playable{

    public void play(){

        System.out.println("playing a volleyball");

    }

}

class BasketBall implements Playable{

    public void play(){

        System.out.println("playing a basketball");

    }

}

class Main{

    public static void main(String[] args){

        System.out.println("Name: archith rollno:av.sc.u4cse24223 class:cse-c ");

        Football football = new Football();

        Volleyball volleyball = new Volleyball();

        BasketBall basketball = new BasketBall();

        football.play();

        volleyball.play();

        basketball.play();

    }

}

Output:

A black screen with white text

AI-generated content may be incorrect.

ERROR TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| s.no | Error | Reason of error | rectification |
| 1 | basketball | Typo error mistake in calling an object | Correcting the typo mistake. |

IMPORTANT POINTS:

1.In this code the main playable is an interface class .By default the method in the interface

abstract.

2.we use the key word implements that is used to access the Playable interface in a class.

3.We are creating objects for each class for calling the methods.