

23CSE111
OBJECT-ORIENTED
PROGRAMMING
LAB MANUAL

Department of computer science and
engineering
Amrita School of Engineering
Amrita Vishwa Vidyapeetham, Amaravati
Campus

Name: Jahnavi

Roll No: 24035

Class: CSE-A

Semester: 2

Verified By:

Date of Submission:

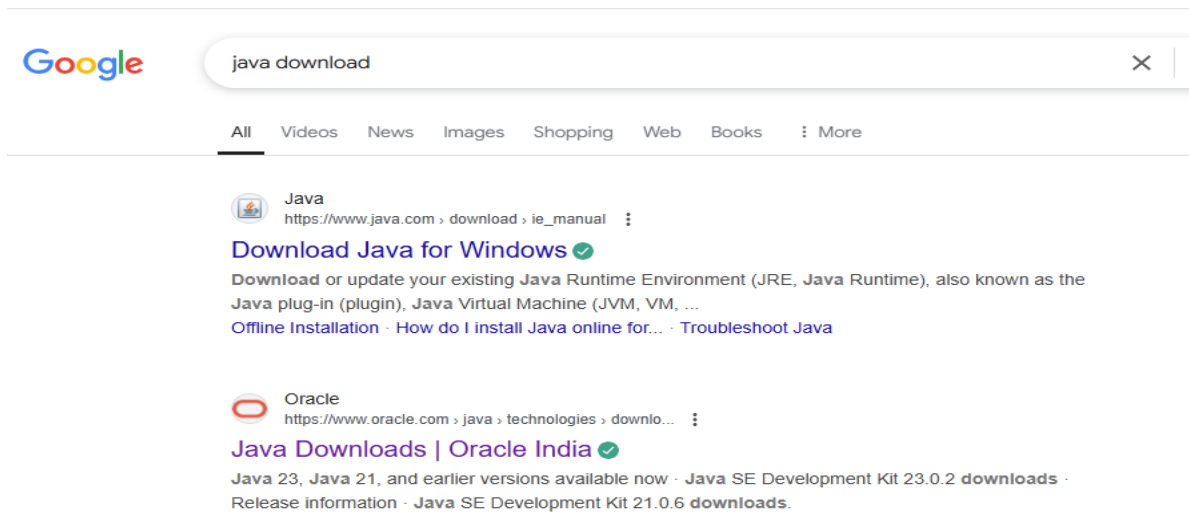
WEEK 1

Program 1

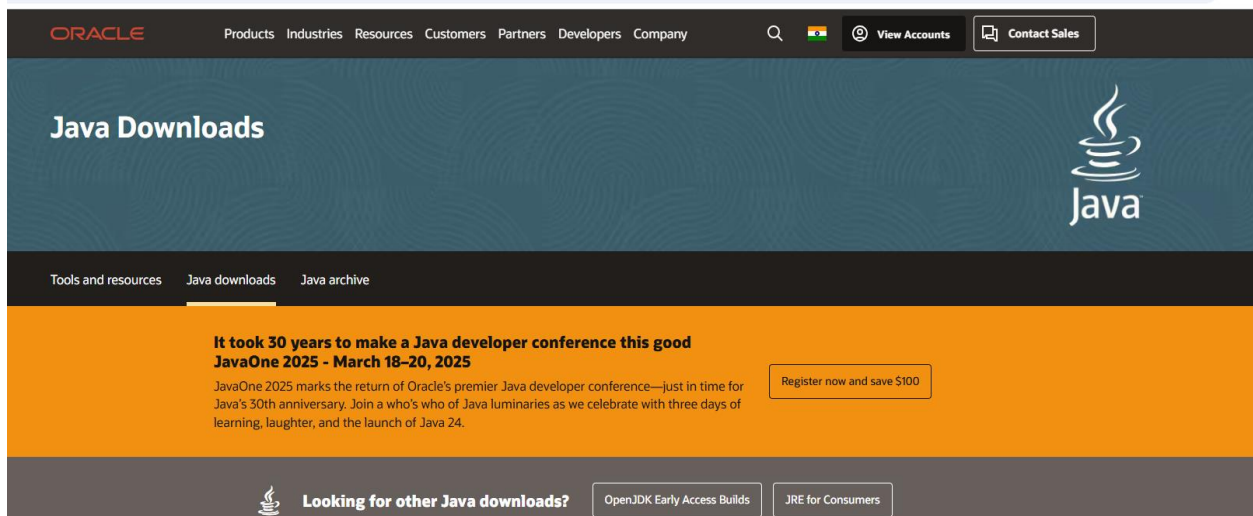
AIM: Downloading and Installing JAVA software

Procedure:

Step 1: Open Google and type java download



Step 2: open website linked with oracle



Step 3: click on JDK 21 and click on windows

Java 23, Java 21, and earlier versions available now

JDK 23 is the latest release of the Java SE Platform.

[Learn about Java SE Subscription](#)

JDK 21 is the latest *Long-Term Support (LTS)* release of the Java SE Platform.

Earlier JDK versions are available below.

JDK 23 **JDK 21** **GraalVM for JDK 23** **GraalVM for JDK 21**

JDK Development Kit 21.0.6 downloads

JDK 21 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions \(NFTC\)](#).

JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the [Java](#) (OTN) and production use beyond the [limited free grants](#) of the OTN license will [require a fee](#).

Linux **macOS** **Windows**

Product/file description	File size	Download
x64 Compressed Archive	185.92 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip (sha256)
x64 Installer	164.31 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe (sha256)
x64 MSI Installer	163.06 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi (sha256)

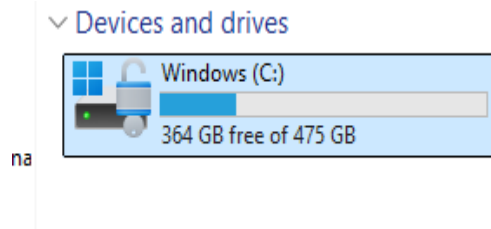
Step 4: After that click on x64 installer

Step 5: the downloading starts

Step 6: After downloading the next step is Installation starts

Step 7: After installation we need to set a path

Step 8: Go to “C” drive in C drive we must select the program files



Name	Date modified	Type	Size
PerfLogs	5/7/2022 10:54 AM	File folder	
Program Files	1/28/2025 6:12 PM	File folder	
Program Files (x86)	9/26/2024 1:54 PM	File folder	
SWSetup	8/31/2024 10:54 AM	File folder	
TC	7/21/2024 8:32 AM	File folder	
Temp	8/2/2024 2:50 PM	File folder	
Users	7/5/2024 7:20 PM	File folder	
Windows	1/27/2025 8:20 PM	File folder	
YOGISOFT	7/21/2024 8:34 AM	File folder	
dosbox-0.74-3-installer_AZR-1K1	7/22/2024 2:28 PM	Application	1,731 KB
tc	7/22/2024 1:46 PM	Application	1,730 KB
turboc	7/22/2024 1:47 PM	Application	1,459 KB

Step 9: Double click on the folder and double click on the java

Avast Software	1/24/2025 4:13 PM	File folder
Common Files	1/22/2025 8:07 PM	File folder
Google	7/5/2024 7:51 PM	File folder
HP	10/4/2024 2:08 PM	File folder
HPCommRecovery	12/15/2023 7:09 PM	File folder
Internet Explorer	10/27/2024 4:58 PM	File folder
Java	1/28/2025 6:31 PM	File folder
MATLAB	10/4/2024 11:04 AM	File folder
McAfee	7/5/2024 7:31 PM	File folder

Name	Date modified	Type	Size
jdk-21	1/28/2025 6:31 PM	File folder	
jdk-23	1/21/2025 9:37 AM	File folder	

Step 10: click on jdk-21 and select bin

bin	1/28/2025 6:31 PM	File folder
conf	1/28/2025 6:31 PM	File folder
include	1/28/2025 6:31 PM	File folder

Step 11: copy the link and paste it in environment

Step 12: select the system variables and select the path

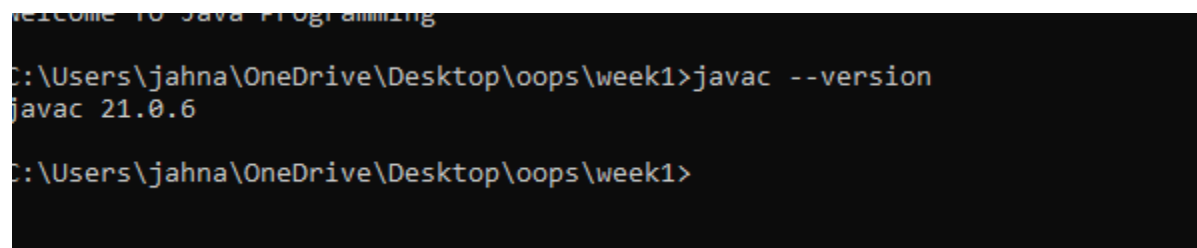
Step 13: after selecting the path we need select new one and need to paste the link

Step 14: after pasting the link we need to click the ok and the apply

Step 15: in this way the downloading and installation of java takes place

Step 16: After installation go to command prompt and type javac – version

Step 17: after we will get the version of the java

A screenshot of a Windows command prompt window. The title bar at the top reads "Welcome To Java Programming". The command prompt shows the directory path "C:\Users\jahna\OneDrive\Desktop\oops\week1" followed by the command "javac --version". The output of the command is "javac 21.0.6". The prompt then returns to "C:\Users\jahna\OneDrive\Desktop\oops\week1>".

```
Welcome To Java Programming
C:\Users\jahna\OneDrive\Desktop\oops\week1>javac --version
javac 21.0.6
C:\Users\jahna\OneDrive\Desktop\oops\week1>
```

Program 2:

AIM: Write a java program to print the message “Welcome To Java Programming”

CODE:

```
class Example1{
    Public static void main(String[] args){
        System.out.println(“Welcome To Java Programming”);
    }
}
```

Output:

```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jahna\OneDrive\Desktop\oops\week1>javac Example1.java

C:\Users\jahna\OneDrive\Desktop\oops\week1>java Example1
Welcome To Java Programming

C:\Users\jahna\OneDrive\Desktop\oops\week1>_
```

Program 3:

AIM: write a java code for printing student name, roll no, class, section

Code:

```
class Example1 {
    public static void main(String[] args) {
        System.out.println("name: Jahnavi");
        System.out.println("roll no: 24035");
        System.out.println("class: CSE");
        System.out.println("Section: CSEA");
    }
}
```

Output:

```
C:\Users\jahna\OneDrive\Desktop\oops\week1>javac Example1.java

C:\Users\jahna\OneDrive\Desktop\oops\week1>java Example1
name: Jahnavi
roll no: 24035
class: CSE
Section: CSEA

C:\Users\jahna\OneDrive\Desktop\oops\week1>_
```

WEEK-2

Program-2

Aim: Write a java program to calculate the area of rectangle

Input:

```
class Areaofrectangle{
    public static void main(String[] args){
        int l=5;
        float b=(float)6.5;
        System.out.println("Area of rectangle="+l*b);
    }
}
```

Output:

```
C:\Users\jahna\OneDrive\Desktop\oops\week1>javac Areaofrectangle.java
C:\Users\jahna\OneDrive\Desktop\oops\week1>java Areaofrectangle
Area of rectangle=32.5
```

Errors:

s.no	Error	Correction
1)	Missing semicolon	Keeping semicolon
2)	End of the flower braces	Closing flower braces

Aim: Write a java program to calculate the simple interest

Input:

```

import java.util.Scanner;

public class SimpleInterest {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        |
        // Input Principal amount
        System.out.print("Enter Principal amount: ");
        double principal = scanner.nextDouble();

        // Input Rate of Interest
        System.out.print("Enter Rate of Interest (per annum): ");
        double rate = scanner.nextDouble();

        // Input Time period in years
        System.out.print("Enter Time period (in years): ");
        double time = scanner.nextDouble();

        // Calculate Simple Interest
        double simpleInterest = (principal * rate * time) / 100;

        // Display the result
        System.out.println("Simple Interest: " + simpleInterest);

        scanner.close();
    }
}

```

Output:

```

C:\Users\jahna\OneDrive\Desktop\oops\week1>javac SimpleInterest.java

C:\Users\jahna\OneDrive\Desktop\oops\week1>java SimpleInterest
Enter Principal amount: 5
Enter Rate of Interest (per annum): 600
Enter Time period (in years): 2
Simple Interest: 60.0

```

Errors:

S.no	Error	Correction
1)	';' excepted Scanner.close()	Add ";" Scanner.close();

Aim: Write a java program to convert temperature from degree farhenit To degree celusis

Input:


```

import java.util.Scanner;
class farhenit{
public static void main(String[]args){

Scanner input=new Scanner(System.in);
System.out.println("enter the temperature in celsius");
double celsius=input.nextDouble();
double farhenit=(celsius*9/5)+32;
System.out.println("temperatute in farhenit:"+farhenit);

}
}

```

Output:

```

C:\Users\jahna\OneDrive\Desktop\oops\week1>javac farhenit.java

C:\Users\jahna\OneDrive\Desktop\oops\week1>java farhenit
enter the temperature in celsius
56
temperatute in farhenit:132.8

```

Errors:

S.no	Error	Correction
1)	error: cannot find symbol fh=(tp*9/5)+32;	Declare the variable: double fh=(tp*9/5)+32;

Aim: Write a program to convert temperature from degree celusis To degree farhenit

Input:

```

import java.util.Scanner;
class celsius{
public static void main(String[]args){

Scanner input=new Scanner(System.in);
System.out.println("enter the temperature in fahrenheit");
double fahrenheit=input.nextDouble();
double celsius=(fahrenheit-32)*5/9;
System.out.println("temperatute in celsius:"+celsius);

}
}

```

Output:

```
C:\Users\jahna\OneDrive\Desktop\oops\week1>javac celsius.java

C:\Users\jahna\OneDrive\Desktop\oops\week1>java celsius
enter the temperature in fahrenheit
78
temperatute in celsius:25.555555555555557
```

Aim: write a java program for factorial number

Input:

```
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: ");
        int number = scanner.nextInt();
        long factorial = 1;

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

        System.out.println("Factorial of " + number + " is " + factorial);
    }
}
```

Output:

```
C:\Users\jahna\OneDrive\Desktop\oops\week1>javac Factorial.java

C:\Users\jahna\OneDrive\Desktop\oops\week1>java Factorial
Enter a number: 6
Factorial of 6 is 720
```

Errors:

S.no	Error	Correction
1)	',' excepted factorial*=i	Add “;” factorial*=i;

Aim: write a java program for largest number

Input:

```
import java.util.Scanner;
class LargestNumber{
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("enter a: ");
        int a = input.nextInt();
        System.out.println("enter b: ");
        int b = input.nextInt();
        System.out.println("enter c: ");
        int c = input.nextInt();
        int largest = (a >= b) ? ((a >= c) ? a : c) : ((b >= c) ? b : c);
        System.out.println("Largest Number: " + largest);
    }
}
```

Output:

```
C:\Users\jahna\OneDrive\Desktop\oops\week1>javac LargestNumber.java
C:\Users\jahna\OneDrive\Desktop\oops\week1>java LargestNumber
enter a:
56
enter b:
45
enter c:
90
Largest Number: 90
C:\Users\jahna\OneDrive\Desktop\oops\week1>
```

Errors:

S.no	Error	Correction
1)	',' excepted int c=input.nextInt()	Add “;” int c=input.nextInt();

WEEK-3

Program-3

Aim: Create the java program with the following instructions.

1. **Create a class with name car**
2. **Create four attributes named car-color, car-brand, fuel-type, milage.**
3. **Create three methods named start, stop, service.**
4. **Create three objects named car1, car2, car3**

Class diagram:

Car
-car_color: String -car_brand: String -car_fuel_type: String -car_milage: String
+Start(): void +Stop(): void

Input:

```

class Car{
private String car_color;
private String car_brand;
private String car_fuel_type;
private String car_mileage;
public void start()
{
System.out.println("Car started");
}
public void stop()
{
System.out.println("car stopped");
}
public Car()
{
System.out.println("welcome to car garage");
}
public static void main(String[]args){
Car car1=new Car();
car1.car_color ="red";
car1.car_fuel_type="petrol";
car1.car_brand="Honda";

Car car2=new Car();
car2.car_color ="Blue";
car2.car_fuel_type="Diesel";
car2.car_brand="Toyato";

Car car3=new Car();
car3.car_color ="white";
car3.car_fuel_type="Electric";
car3.car_brand="FORD";
car1.start();
car1.stop();

car2.start();
car2.stop();

car3.start();
car3.stop();

System.out.println("car1_color: "+car1.car_color+" brand:"+car1.car_brand+"fuel_type:"+car1.car_fuel_type);
System.out.println("car2_color: "+car2.car_color+" brand:"+car3.car_brand+"fuel_type:"+car2.car_fuel_type);
System.out.println("car3_color: "+car3.car_color+" brand:"+car3.car_brand+"fuel_type:"+car3.car_fuel_type);

}
}

```

Output:

```

C:\Users\jahna\OneDrive\Desktop\oops\week1>javac Car.java

C:\Users\jahna\OneDrive\Desktop\oops\week1>java Car
welcome to car garage
welcome to car garage
welcome to car garage
Car started
car stopped
Car started
car stopped
Car started
car stopped
car1_color: red brand:Hondafuel_type:petrol
car2_color: Blue brand:FORDfuel_type:Diesel
car3_color: white brand:FORDfuel_type:Electric

```

Errors:

S.no	Error	Correction
1)	illegal start of type Public void stop{	Add “)” Public void stop(){
2)	Cannot find symbol thiscar_brand=car_brand	Add “.” this.car_brand=car _brand

Aim: Write a java program for Bank account by using constructor and for withdrawal and deposit .

Class diagram:

BankAccount
-Name: String -AccNo: String -currbal : String
+withdraw(int WAmt): int +deposit(int DAmt): int

Input:

```

class BankAccount {
    String Name;
    int AccNO, currbal;

    BankAccount(String Name, int AccNO, int currbal) {
        this.Name = Name;
        this.AccNO = AccNO;
        this.currbal = currbal;
        System.out.println("The customer details are: " + this.Name + " " + this.AccNO + " " + this.currbal);
    }

    public void withdraw(int withdrawalAmount) {
        if (withdrawalAmount <= currbal) {
            currbal = currbal - withdrawalAmount;
            System.out.println("Remaining balance: " + currbal);
        } else {
            System.out.println("Insufficient funds");
        }
    }

    public int deposit(int depositAmount) {
        currbal = currbal + depositAmount;
        return currbal;
    }

    public static void main(String[] args) {
        BankAccount ram = new BankAccount("ram", 1234, 1000);
        ram.withdraw(500); // Example of valid withdrawal
        ram.withdraw(500); // Another example of valid withdrawal
        int finalAmount = ram.deposit(5001);
        System.out.println("Final balance: " + finalAmount);
    }
}

```

Output:

```

C:\Users\jahna\OneDrive\Desktop\oops\week1>javac BankAccount.java

C:\Users\jahna\OneDrive\Desktop\oops\week1>java BankAccount
The customer details are: ram 1234 1000
Remaining balance: 500
Remaining balance: 0
Final balance: 5001

```

Errors:

S.no	Errors	Correction
1)	; excepted Ram.withdraw(500)	Add ";" Ram.withdraw(500);
2)	Cannot find symbol thisCurrBal=CurrBal	Add "." this.CurrBal=CurrBal

WEEK-4

Program-1

Aim: Write java program with class name Book. This class should contain various attributes such as title of the book, author, year of publication. It should also contain constructor with parameter which initializes the title of the book, author, year of publication. create a method which displays the details of the book that is title of the book, author, year of publication display details of two books by creating two objects.

Class Diagram:

Book
-title: String -author: String -yearOfPublication: int
+Book(): void +display() : void

Input:

```
public class Book {
    private String title;
    private String author;
    private int yearOfPublication;

    public Book(String title, String author, int yearOfPublication) {
        this.title = title;
        this.author = author;
        this.yearOfPublication = yearOfPublication;
    }

    public void displayDetails() {
        System.out.println("Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("Year of Publication: " + yearOfPublication);
    }

    public static void main(String[] args) {
        Book book1 = new Book("To Kill a Mockingbird", "Harper Lee", 1960);
        Book book2 = new Book("1984", "George Orwell", 1949);

        System.out.println("Book 1 Details:");
        book1.displayDetails();

        System.out.println("\nBook 2 Details:");
        book2.displayDetails();
    }
}
```


Output:

```
C:\Users\jahna\OneDrive\Desktop\oops\week1>javac Book.java
C:\Users\jahna\OneDrive\Desktop\oops\week1>java Book
Book 1 Details:
Title: To Kill a Mockingbird
Author: Harper Lee
Year of Publication: 1960
Book 2 Details:
Title: 1984
Author: George Orwell
Year of Publication: 1949
```

Errors:

s.no	Error	Correction
1)	Illegal start of type public void display({	Add “)” public void dispaly(){
2)	Cannot find symbol thisauthor=author;	Add “.” this.author=author;

Program-2

Aim: To create a java class named MyClass with astatic variable "count" of int type initialized to zero and a constant variable pi of type double initialized to 3.1415 as attributes of that class. define a constructor for myclass that increments the count variable each time an object of my class is created. finally print the final values of "count" and "pi"

Class diagram:

MyClass
+count: int +pi : double

Input:

```

public class MyClass {
    public static int count = 0;
    public static final double pi = 3.1415;

    public MyClass() {
        count++;
    }

    public static void main(String[] args) {
        MyClass obj1 = new MyClass();
        MyClass obj2 = new MyClass();
        MyClass obj3 = new MyClass();

        System.out.println("Final Count: " + count);
        System.out.println("Value of Pi: " + pi);
    }
}

```

Output:

```

C:\Users\jahna\OneDrive\Desktop\oops\week1>javac MyClass.java

C:\Users\jahna\OneDrive\Desktop\oops\week1>java MyClass
Final Count: 3
Value of Pi: 3.1415

```

Errors:

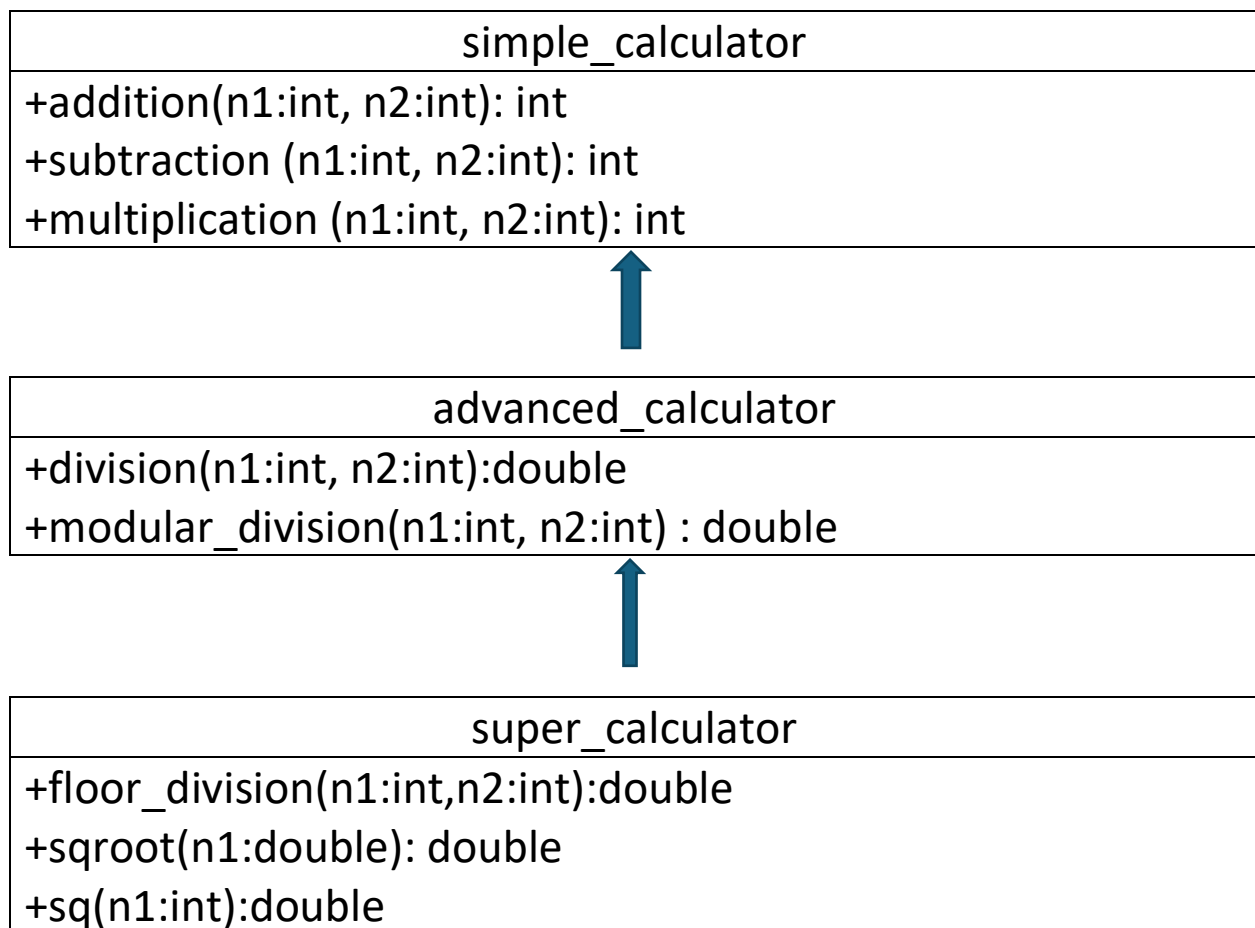
S.no	Error	Correction
1)	;expected MyClass()	Add ; MyClass():
2)	Missing Closing Syntax }	Keeping Closing Syntax }

Week-5

Program-1

Aim: Create a calculator using the operations including addition, subtraction, multiplication, division, square, square root, cube, floor using multiple inheritance and display desired output

Class Diagram:



Input:

```

import java.util.Scanner;

class simple_calculator {
    // Methods
    public int addition(int n1, int n2) {
        return n1 + n2;
    }

    public int subtraction(int n1, int n2) {
        return n1 - n2;
    }

    public int multiplication(int n1, int n2) {
        return n1 * n2;
    }
}

class advanced_calculator extends simple_calculator {
    // Methods
    public double division(int n1, int n2) {
        if (n2 != 0) {
            return (double) n1 / n2; // Cast to double to ensure correct division
        } else {
            System.out.println("Division with 0 is not possible.");
            return Double.NaN; // Return NaN for invalid division
        }
    }

    public double modular_division(int n1, int n2) {
        if (n2 != 0) {
            return n1 % n2;
        } else {
            System.out.println("Modular division with 0 is not possible.");
            return Double.NaN; // Return NaN for invalid modulo operation
        }
    }
}

// Derived class from advanced_calculator
class super_calculator extends advanced_calculator {
    // Methods
    public int floor_division(int n1, int n2) {
        if (n2 != 0) {
            return Math.floorDiv(n1, n2); // Proper floor division
        } else {
            System.out.println("Floor division with 0 is not possible.");
            return Integer.MIN_VALUE; // Return a placeholder for invalid operation
        }
    }

    public double sqroot(double n1) {
        if (n1 >= 0) {
            return Math.sqrt(n1);
        } else {
            System.out.println("Square root of negative number is not possible.");
            return Double.NaN; // Return NaN for invalid square root
        }
    }

    public double sq(int n1) {
        return n1 * n1;
    }
}

```

```
// Main class
public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the first number: ");
        int a = input.nextInt();
        System.out.print("Enter the second number: ");
        int b = input.nextInt();

        super_calculator sc = new super_calculator();
        System.out.println("Addition: " + sc.addition(a, b));
        System.out.println("Subtraction: " + sc.subtraction(a, b));
        System.out.println("Multiplication: " + sc.multiplication(a, b));
        System.out.println("Division: " + sc.division(a, b));
        System.out.println("Modular division: " + sc.modular_division(a, b));
        System.out.println("Floor Division: " + sc.floor_division(a, b));
        System.out.println("Square Root: " + sc.sqroot(a));
        System.out.println("Square: " + sc.sq(a));

        input.close(); // Fixed missing semicolon
    }
}
```

Output:

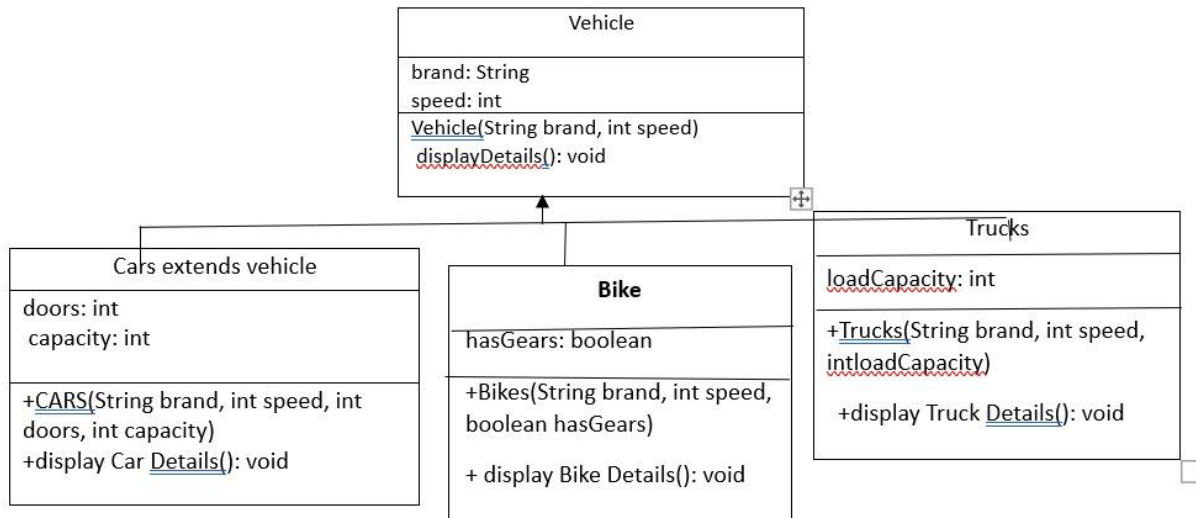
Error:

S.no	Error	Correction
1)	‘;’ excepted tv.v_final()	Add “;” tv.v_final();
2)	illegal start of type public void v_final({	Add “)” public void v_final(){

Program:2

Aim: A vehicle enter company wants to develop his system that maintains, information about different types of vehicles available for rent, the company rents out cars and boke and they need a program to store details about each vehicle such as brand and speed

Class diagram



Input:

```
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:

```
C:\Users\jahna\OneDrive\Desktop\oops\week1>javac Vehicle.java

C:\Users\jahna\OneDrive\Desktop\oops\week1>java Rent
Number of doors:5
Capacity:5
Brand:Tayota
Speed:120
This bike has gears.
Brand:KTM
Speed:80
The capacity of truck is: 1
Brand:TATA
Speed:100
```

Errors:

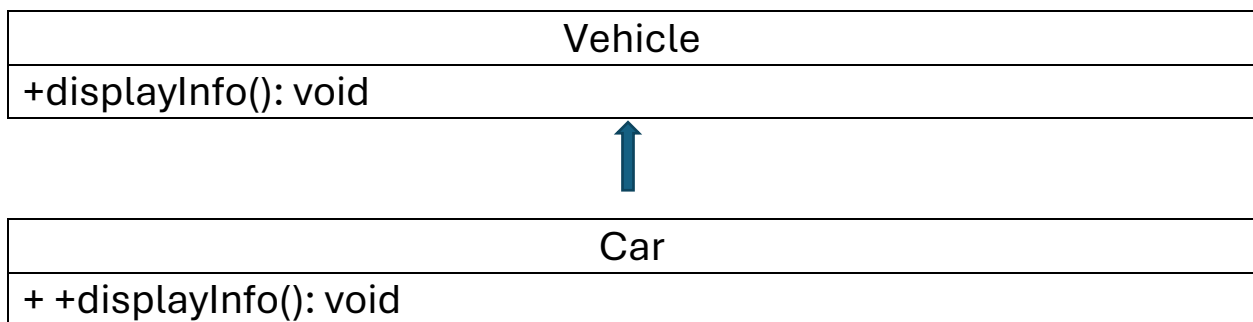
S.no	Error	Correction
1)	Cannot find symbol thisbrand=brand;	Add "." this.brand=brand;
2)	Missing closing syntax }	Keeping closing syntax }

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

Class Diagram:



Input:

```
class Vehicle {
    public void displayInfo() {
        System.out.println("This is a vehicle.");
    }
}

class Car extends Vehicle {
    @Override
    public void displayInfo() {
        System.out.println("This is a car. It has 4 wheels and runs on fuel.");
    }
}

class Main{
    public static void main(String[] args) {
        Vehicle myVehicle = new Vehicle();
        myVehicle.displayInfo(); // Calls Vehicle's method

        Car myCar = new Car();
        myCar.displayInfo(); // Calls Car's overridden method
    }
}
```

Output:


```
C:\Users\jahna\OneDrive\Desktop\oops\week1>javac Vehicle.java
C:\Users\jahna\OneDrive\Desktop\oops\week1>java Main
This is a vehicle.
This is a car. It has 4 wheels and runs on fuel.
```

Errors:

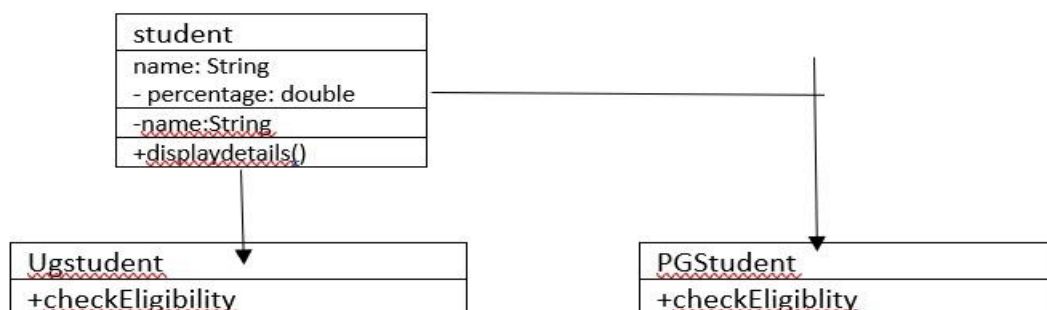
S.no	Error	Correction
1)	package Sysetm does not exist Sysetm.out.println	Chaning name as System.out.println
2)	cannot find symbol vechile extends Students	Correcting the file name as Vehicle

Program-2:

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

- UG admission require a min of 60%
- PG admission require a min of 70%

class diagram



Input:

```

class Student{
String name;
double percentage;
public Student(String name, double percentage){
this.name=name;
this.percentage=percentage;
System.out.println(name+" should meet admission member");
}
}
class Ug extends Student{
public Ug(String name, double percentage){
super(name, percentage);
if(percentage>=60)
{
System.out.println(name + "eligible for Ug admission");
}
else{
System.out.println(name + " not eligible for Ug admission");
}
}
}
class Pg extends Student{
public Pg(String name, double percentage){
super(name, percentage);
if(percentage>=70)
{
System.out.println(name + "eligible for Pg admission");
}
else{
System.out.println(name + " not eligible for Pg admission");
}
}
}
class Eligibility{
public static void main(String[] args){
Ug u= new Ug("Janu", 76);
Ug g= new Ug("mokshu", 54);
Pg p= new Pg("Krish", 89);
}
}
}

```

Output:

```

C:\Users\jahna\OneDrive\Desktop\oops\week1>javac Student.java
C:\Users\jahna\OneDrive\Desktop\oops\week1>java Eligibility
Janu should meet admission member
Janueligible for Ug admission
mokshu should meet admission member
mokshu not eligible for Ug admission
Krish should meet admission member
Krisheligible for Pg admission

```

Error:

S.no	Error	Correction
1)	package Sysetm does not exist Sysetm.out.println	Chaning name as System.out.println
2)	cannot find symbol Pg extends Students	Removing "s" from students

Program-3

Aim: Create a class with overloaded method to perform addition:

- i. Add two integers
- ii. Add two doubles
- iii. Add three integers

Class Diagram:

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

Input:

```
class Calculator{
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 Main{
public static void main(String[] args){
Calculator calc= new Calculator();
System.out.println("The sum of 5 and 3 is:"+calc.add(5,3));
System.out.println("The sum of 5 and 3 is:"+calc.add(5,3));
System.out.println("The sum of 5,3 and 4 is:"+calc.add(5,3,4));
}
}
```

Output:

```
C:\Users\jahna\OneDrive\Desktop\oops\week1>javac Calculator.java

C:\Users\jahna\OneDrive\Desktop\oops\week1>java Main
The sum of 5 and 3 is:8
The sum of 5 and 3 is:8
The sum of 5,3 and 4 is:12
```

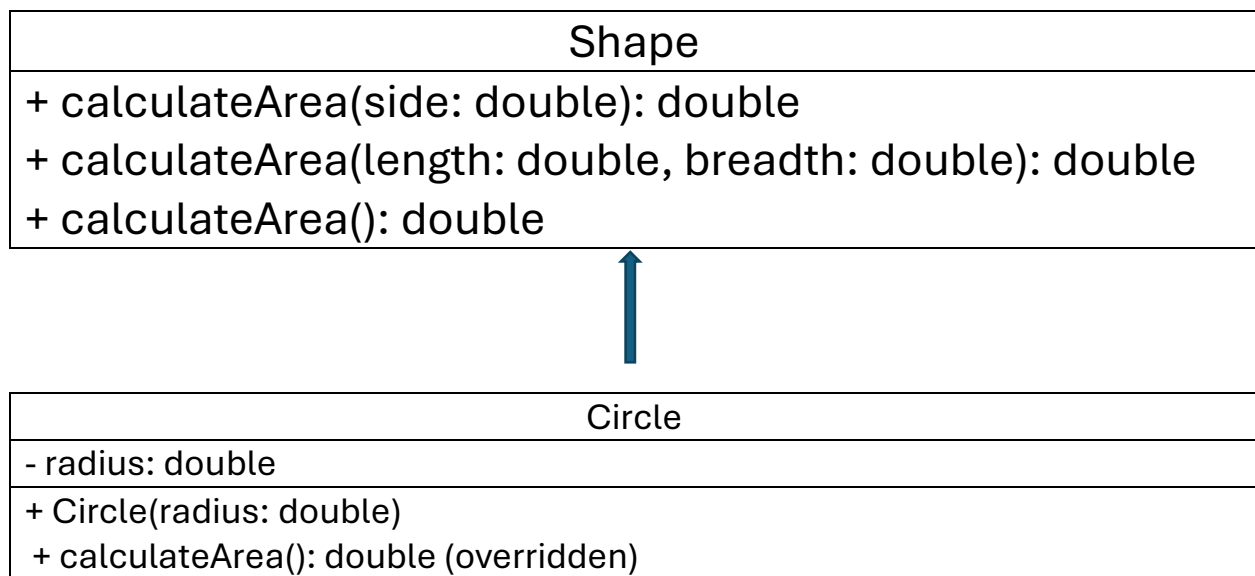
Errors:

S.no	Error	Correction
1)	package Sysetm does not exist Sysetm.out.println	Chaning name as System.out.println
2)	Missing closing syntax }	Keeping closing syntax }

Program-4

AIM: Create a shape class with a method calculate area(), that is overloaded for different shape then create a subclass circle that overrides calculate the area() method for a circle

Class Diagram:



Input:

```

public class Shape {
    // Overloaded calculateArea() for a square
    public double calculateArea(double side) {
        return side * side;
    }
    // Overloaded calculateArea() for a rectangle
    public double calculateArea(double length, double breadth) {
        return length * breadth;
    }
    // This method can be overridden by subclasses
    public double calculateArea() {
        return 0; // Default implementation
    }
    public static void main(String[] args) {
        Shape shape = new Shape();
        System.out.println("Area of square: " + shape.calculateArea(5)); // Square
        System.out.println("Area of rectangle: " + shape.calculateArea(5, 10)); // Rectangle

        // Using Circle subclass
        Circle circle = new Circle(7); // Circle with radius 7
        System.out.println("Area of circle: " + circle.calculateArea()); // Circle
    }
}

// Circle subclass
class Circle extends Shape {
    private double radius;

    // Constructor to initialize radius
    public Circle(double radius) {
        this.radius = radius;
    }

    // Overriding calculateArea() for a circle
    @Override
    public double calculateArea() {
        return Math.PI * radius * radius;
    }
}

```

Output:

```

C:\Users\jahna\OneDrive\Desktop\oops\week1>javac Shape.java

C:\Users\jahna\OneDrive\Desktop\oops\week1>java Shape
Area of square: 25.0
Area of rectangle: 50.0
Area of circle: 153.93804002589985

```

Error:

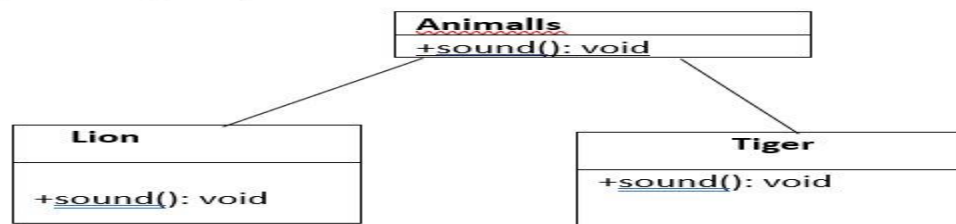
S.no	Error	Correction
1	calculateArea() in Shape has no proper implementation.	Add a default implementation or make it an abstract method.
2	code is missing comments.	Add short comments to explain the purpose of methods and classes.

Week-7

Program-1

Aim: 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 sepific sound for each animal

Class diagram:



Input:

```
// Abstract class Animal
abstract class Animal {
    // Abstract method
    abstract void sound();
}

// Subclass Lion that extends Animal
class Lion extends Animal {
    @Override
    void sound() {
        System.out.println("Lion roars!");
    }
}

// Subclass Tiger that extends Animal
class Tiger extends Animal {
    @Override
    void sound() {
        System.out.println("Tiger growls!");
    }
}

// Main class to test the program
class AnimalTest {
    public static void main(String[] args) {
        Animal lion = new Lion();
        Animal tiger = new Tiger();

        lion.sound();
        tiger.sound();
    }
}
```

Output:

```
C:\Users\jahna\OneDrive\Desktop\oops\week1>java AnimalTest
Lion roars!
Tiger growls!
```

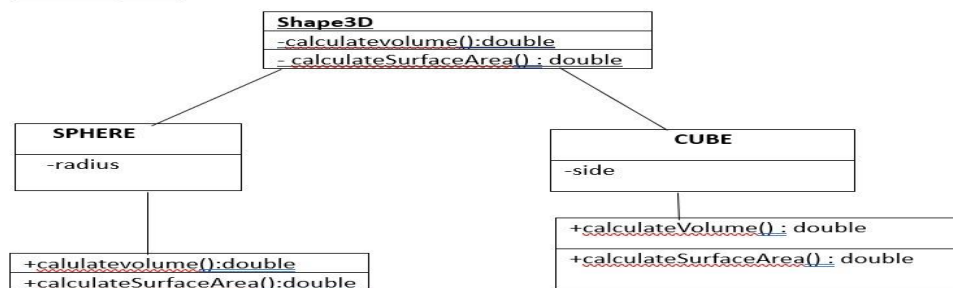
Error:

S.no	Error	Correction
1	Case sensitive	Fixed method name consistency
2	Improper class naming	Renamed Main to AnimalTest for clarity

Program-2

Aim: write a java program to create an abstract class Shape3D with abstract methods calculate Volume() and calculate Surfacearea(). create subclasses sphere and Cube that extend shape3Dclass and implement the respective methods to calculate the volume and surface area of each shape

Class diagram:



Input:

```
abstract class Shape3D {
    abstract double calculateVolume();
    abstract double calculateSurfaceArea();
}
class Sphere extends Shape3D {
    private double radius;
    public Sphere(double 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 {
    private double side;
    public Cube(double side) {
        this.side = side;
    }
    double calculateVolume() {
        return Math.pow(side, 3);
    }
    double calculateSurfaceArea() {
        return 6 * Math.pow(side, 2);
    }
}
class Shape3DTest {
    public static void main(String[] args) {
        Shape3D sphere = new Sphere(5.0); // Radius = 5
        Shape3D cube = new Cube(4.0); // Side = 4

        System.out.println("Sphere Volume: " + sphere.calculateVolume());
        System.out.println("Sphere Surface Area: " + sphere.calculateSurfaceArea());

        System.out.println("Cube Volume: " + cube.calculateVolume());
        System.out.println("Cube Surface Area: " + cube.calculateSurfaceArea());
    }
}
```

Output:

```
C:\Users\jahna\OneDrive\Desktop\oops\week1>java Shape3DTest
Sphere Volume: 523.5987755982989
Sphere Surface Area: 314.1592653589793
Cube Volume: 64.0
Cube Surface Area: 96.0
```

Error:

S.no	Error	Correction
1	System.out.println("Sphere Surface Area: " + sphere.calculateSurfaceArea())	System.out.println("Sphere Surface Area: " + sphere.calculateSurfaceArea());
2	Shape3D shape = new Shape3D();	Shape3D sphere = new Sphere(5);

Program-3

Aim: write a java program using an abstract class to define a method for pattern printing

1.create an abstract class named pattern printer with an abstract method print pattern and a concentre method to display the pattern title

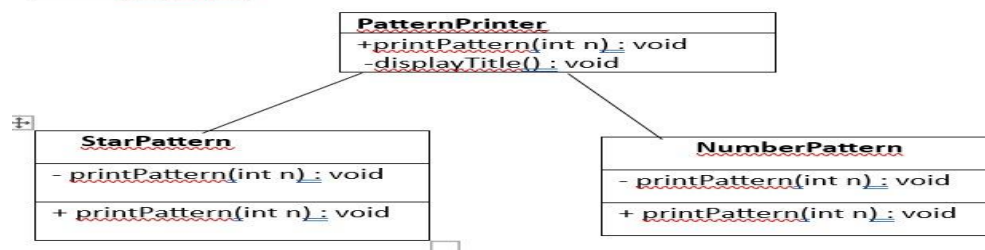
Implement two subclasses

1.star pattern-prints a right angled triangle of *

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

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

Class digram:



Input:


```

abstract class PatternPrinter {
    abstract void printPattern(int n);
    void displayTitle(String title) {
        System.out.println("==== " + title + " =====");
    }
}

class StarPattern extends PatternPrinter {
    @Override
    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
    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 Main2 {
    public static void main(String[] args) {
        int number = 5;
        PatternPrinter star = new StarPattern();
        star.displayTitle("Star Pattern");
        star.printPattern(number);
        System.out.println();
        PatternPrinter numberPat = new NumberPattern();
        numberPat.displayTitle("Number Pattern");
        numberPat.printPattern(number);
    }
}

```

Output:

```

===== Star Pattern =====
*
* *
* * *
* * * *
* * * * *

===== Number Pattern =====
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

```

Error:

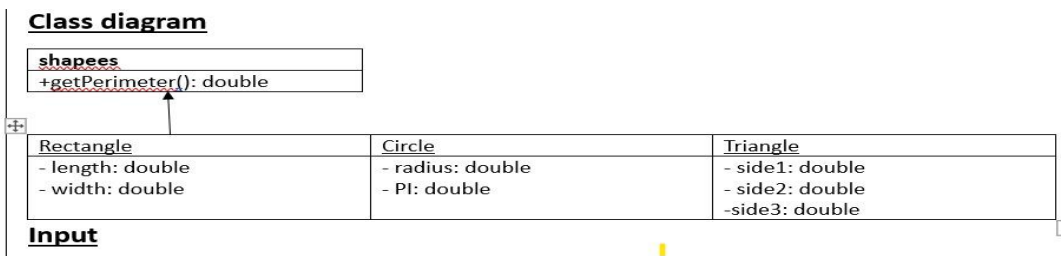
S.no	Error	Correction
1	File name mismatch.	Rename the file to Calculator1.java to match the class name.
2	Case sensitive	Fixed method name consistency

Week-8

Program-1

Aim: write a java program to create an interface shape with getperimeter() method. Create three classes rectangle, circle and triangle the shape interface implement the seet perimeter method for each 3 classes

Class Diagram:



Input:

```
abstract class Shapes {
    public abstract void area();
}
class Triangle extends Shapes {
    double base, height;
    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }
    @Override
    public void area() {
        double result = 0.5 * base * height;
        System.out.println("This is Jahnvi Roll No:24035 Section:CSE-A");
        System.out.println("Area of Triangle: " + result);
    }
}
class Circle extends Shapes {
    double radius;
    public Circle(double radius) {
        this.radius = radius;
    }
    @Override
    public void area() {
        double result = Math.PI * radius * radius;
        System.out.println("Area of Circle: " + result);
    }
}
class Rectangle extends Shapes {
    double length, width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    @Override
    public void area() {
        double result = length * width;
        System.out.println("Area of Rectangle: " + result);
    }
}
class ShapeDemo {
    public static void main(String[] args) {
        Shapes triangle = new Triangle(5, 10);
        Shapes circle = new Circle(7);
        Shapes rectangle = new Rectangle(8, 4);
        triangle.area();
        circle.area();
        rectangle.area();
    }
}
```

Output:

```
C:\Users\jahna\OneDrive\Desktop\oops\week1>java ShapeDemo
This is Jahnvi Roll No:24035 Section:CSE-A
Area of Triangle: 25.0
Area of Circle: 153.93804002589985
Area of Rectangle: 32.0
```

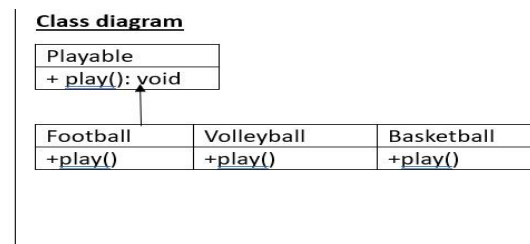
Error:

S.no	Error	Correction
1	File name mismatch.	Rename the file to Calculator1.java to match the class name.
2	Case sensitive	Fixed method name consistency

Program-2

Aim: write a java program to create an interface Playable with a method paly() that takes no arguments and return void create 3 classes football, volleyball, and basketball that implements the playable interface and override the play() method to play respective sports.

Class Diagram:



Input:

```

interface Playable{
void play();
}
class Football implements Playable{
public void play(){
System.out.println("playing football");
}
}
class Volleyball implements Playable{
public void play(){
System.out.println("playing volleyball");
}
}
class Basketball implements Playable{
public void play(){
System.out.println("playing Basketball");
}
}
class Main_v4{
public static void main(String[]args){
Playable football = new Football();
Playable volleyball = new Volleyball();
Playable basketball = new Basketball();
football.play();
volleyball.play();
basketball.play();
}
}

```

Output:

```

playing football
playing volleyball
playing Basketball
PS C:\Users\jahna>

```

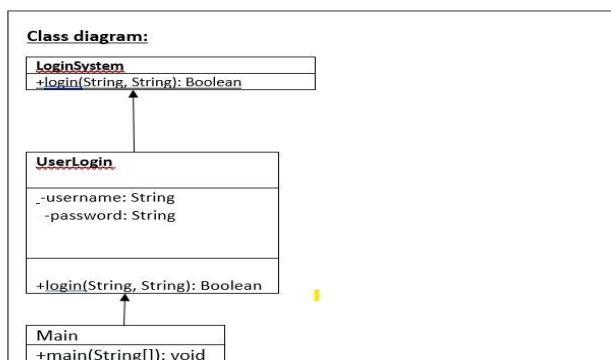
Error:

Error	Rectification	Correction
1	Case senstive	corrected
2	File location error	Replacing with correct location

Program-3

Aim: Write a java program to implement the login system using interfaces

Class Diagram:



Input:

```
import java.util.Scanner;
interface LoginSystem {
    boolean login(String username, String password);
}
class UserLogin implements LoginSystem {
    private final String username = "admin";
    private final String password = "password123";
    public boolean login(String username, String password) {
        return this.username.equals(username) && this.password.equals(password);
    }
}
class Main4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        LoginSystem loginSystem = new UserLogin();
        System.out.println(" Login System ");
        System.out.print("Enter Username: ");
        String inputUsername = scanner.nextLine();
        System.out.print("Enter Password: ");
        String inputPassword = scanner.nextLine();
        if (loginSystem.login(inputUsername, inputPassword)) {
            System.out.println("Login successful! Welcome.");
        } else {
            System.out.println("Login failed! Invalid credentials.");
        }
        scanner.close();
    }
}
```

Output:

```
C:\Users\jahna\OneDrive\Desktop\oops\week1>java Main4
 Login System
Enter Username: janu
Enter Password: 2006
Login failed! Invalid credentials.
```

Error:

S.no	Error	Correction
1	Case sensitive	Fixed method name consistency
2	Improper class naming	Renamed Main to Main4 for clarity