23CSE111

OBJECT ORIENTED PROGRAMMING

LAB MANUAL

Department of Computer and Science Engineering Amrita

School of Engineering

Amrita Vishwa Vidyapeetham, Amaravati Campus

Name: Vanshika Chiratanagalla

Roll No: CSE24038

Verified By:

S. No	Programs	Date	Page No	Signature
WEEK 1		27-01-2025		
1	Write the steps to download and install Java.			
2	Write a java program to print the message “Welcome to java programming”.			
3	Write a java program that prints name, roll number and section of a student.			
WEEK 2		3-02-2025		
1	Write a java program to calculate the area of a rectangle.			
2a)	Write a program to convert temperature from Celsius to Fahrenheit			
b)	Write a program to convert temperature from Fahrenheit to Celsius.			
3	Write a program to calculate the simple interest			
4	Write a program to find the largest of three numbers using ternary operator.			
5	Write a program to find the factorial of a number			
WEEK 3		11-02-2025		
1	Creating a car class with the given instructions			

	Creating a BankAccount class with the given instructions			
WEEK 4		02-03-2025		
1	Write a java program with class named “Book” with given instructions.			
2	To create a java program with class named Myclass with given instructions.			
WEEK 5		09-03-2025		
1	Create a calc using the operations including add, sub, mul, div using multilevel inheritance and display the desired output.			
2	Creating a Rental System.			
WEEK 6		16-03-2025		
1	Write a java program to create a Vehicle class with displayInfo() method , overridden in Car subclass to provide info about carcompany , model , price ,seating and petrol.			
2	An automated admission system that verifies student eligibility for UG and PG with different criteria. 1.UG requires minimum of 60% 2.PG requires minimum of 70%			
3	Create a calculator class with overloaded methods to perform additions 1.add two integers			

	2.add two double values 3.add three integers			
4	Create a shape class with method calculateArea() that is overloaded for different shapes (eg: square, rectangle).Then create a subclass Circle that overrides calculateArea() method for Circle.			
WEEK 7		14-04-2025		
1	Write a java program to create an abstract class Animal with abstract method sound and create subclasses Lion and Tiger that implements the method.			
2	Write a java program to create an abstract class shape3D with abstract methods to calculate volume and surfacearea and create subclasses for sphere and cube that implements these methods.			
3	Create an abstract class PatternPrint with an abstract method printing to print the pattern and a concrete method to display the pattern . Implement the patterns 1) Star Pattern - prints a right angled triangle of stars 2) Number Pattern – prints a right angled triangle of increasing numbers.			

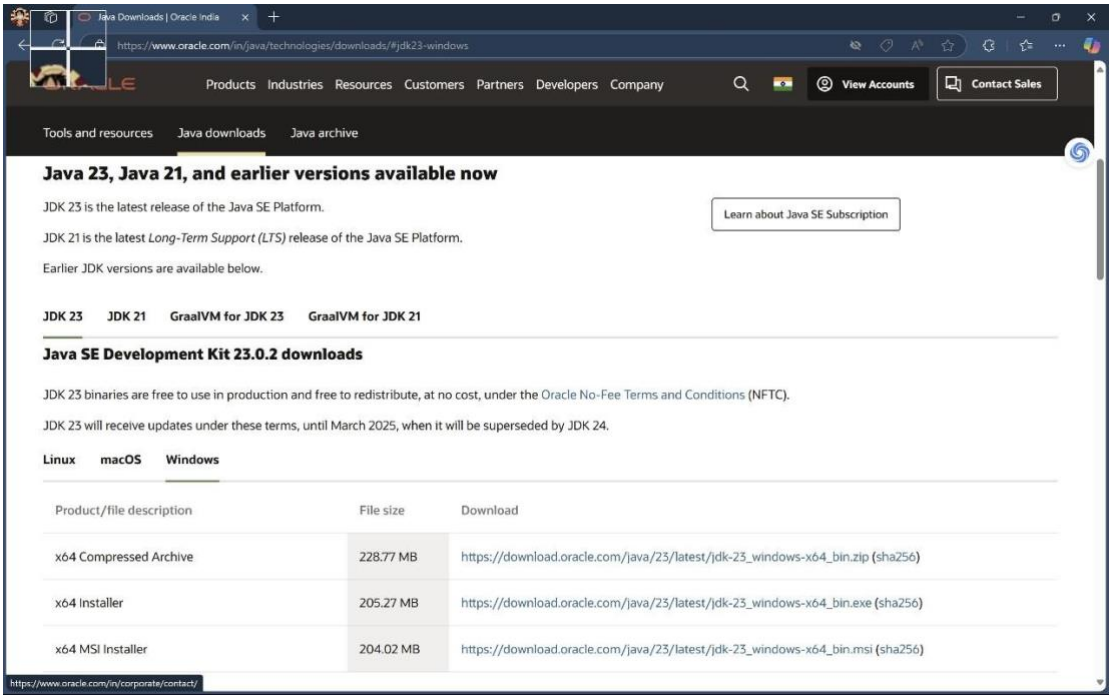
WEEK –1

1. Write the steps to download and install Java.

Aim: To download and install java.

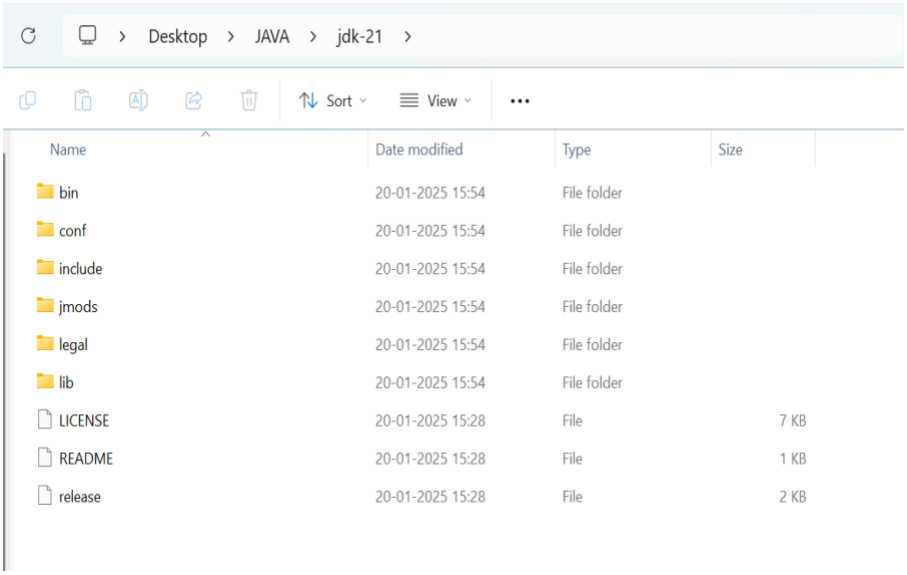
Procedure:

i. Visit oracle.com website to download Java.

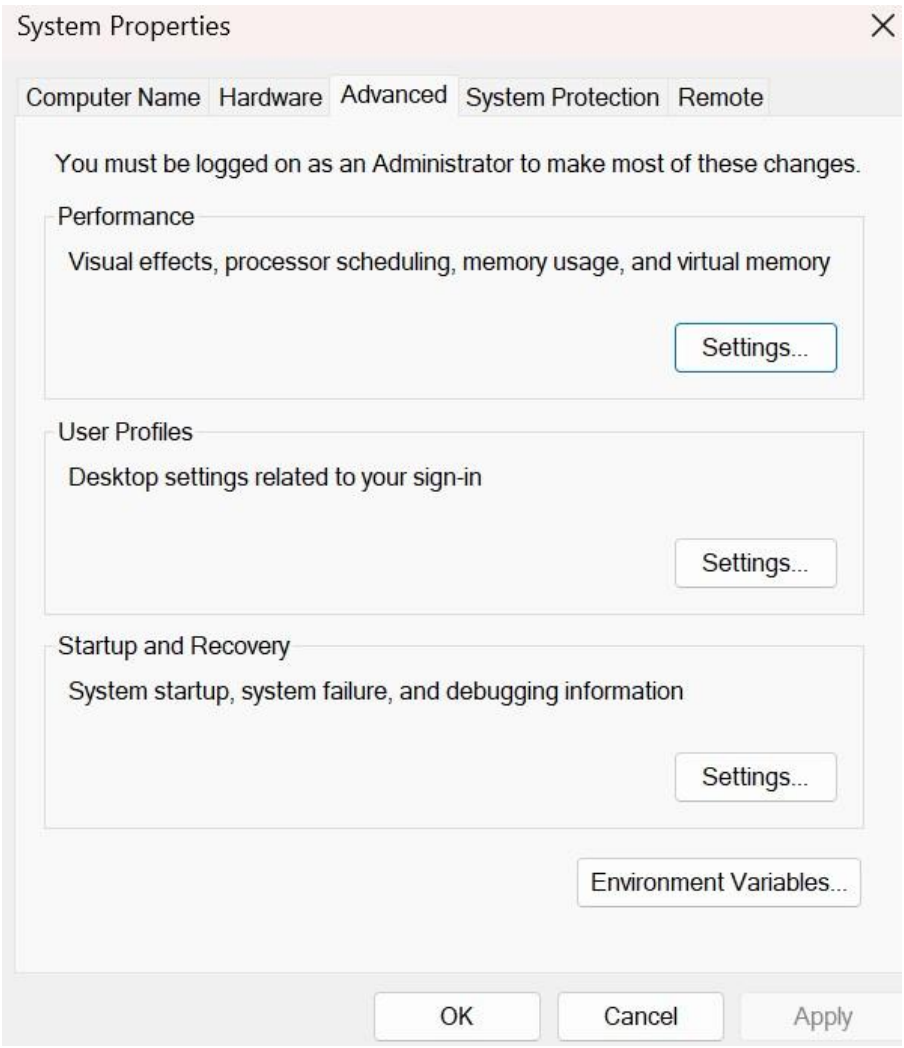


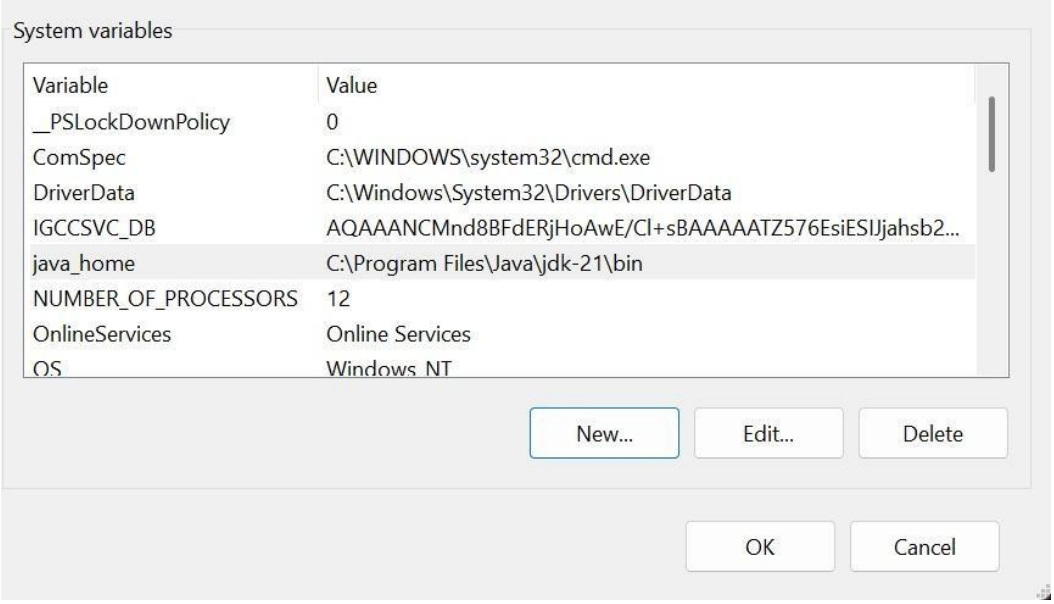
ii. Download the version which supports LTS (JDK 21) x64 installer for windows.

iii. Install and copy the path.



iv. Open environmental variables and add a new file with path.





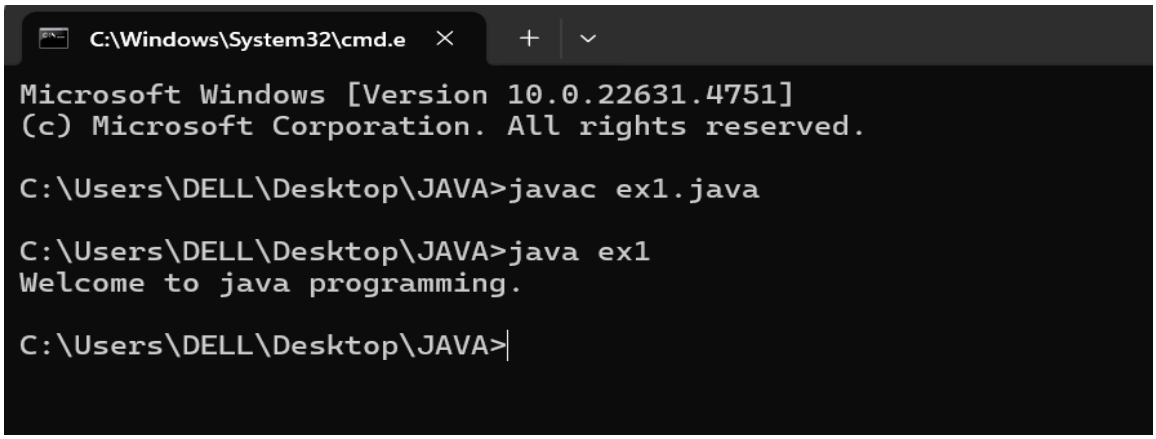
v. Verify java version in command window.

2. Write a java program to print the message “Welcome to java programming”.

Code:

```
class
ex1{
    public static void main(String[] args){
   System.out.println("Welcome to java programming.");
        }
}
```

Output:



Error:

S.No	Expected Error	Reason
1	;	; is expected at end
2	S	Capital S is expected for String and System.

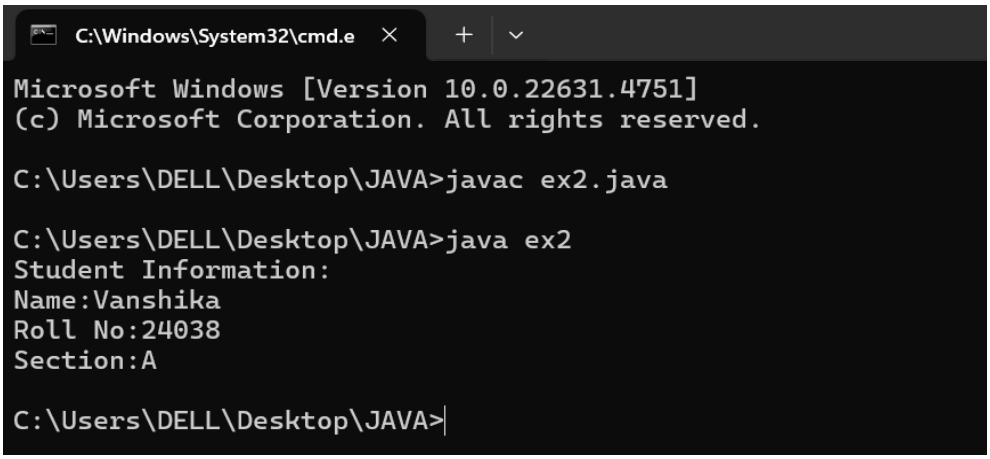
3.Write a java program to print the name, roll number and section of a student.

Code:

```
class
ex2{
        public static void main(String[]
args){          String name =
"Vanshika";
int rollNo = 24038;
```

```
        String section = "A";


       System.out.println("Student Information:");
       System.out.println("Name:" + name);
      System.out.println("Roll No:" + rollNo);
       System.out.println("Section:" + section);
}
}
```

Output:



Error:

S.No	Expected Error	Reason
1	S	Capital S is expected for String and System.

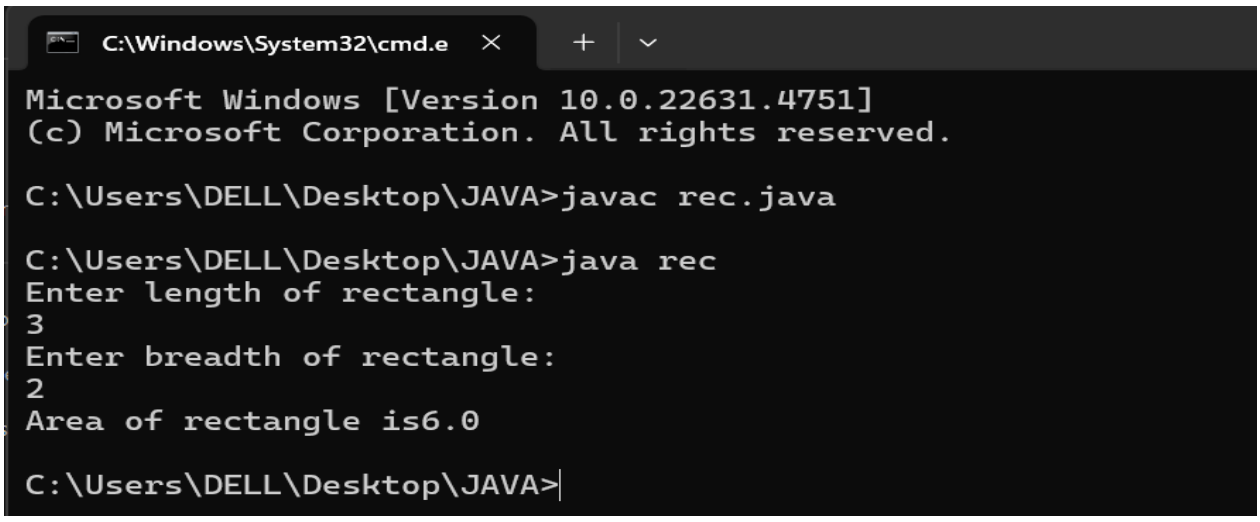
WEEK –2

1. Write a java program to calculate the area of a rectangle.

Code:

```
import java.util.Scanner;
class rec{
	public static void main(String[] args){
				Scanner scan=new Scanner(System.in);
				System.out.println("Enter length of rectangle:");
				double l=scan.nextDouble();
				System.out.println("Enter breadth of rectangle:");
				double b=scan.nextDouble();
				double a=l*b;
				System.out.println("Area of rectangle is"+a);
	}
}
```

Output:



```
C:\Users\DELL\Desktop\JAVA>javac rec.java

C:\Users\DELL\Desktop\JAVA>java rec
Enter length of rectangle:
h
Exception in thread "main" java.util.InputMismatchException
        at java.base/java.util.Scanner.throwFor(Scanner.java:947)
        at java.base/java.util.Scanner.next(Scanner.java:1602)
        at java.base/java.util.Scanner.nextDouble(Scanner.java:2573)
        at rec.main(rec.java:6)

C:\Users\DELL\Desktop\JAVA>
```

Error:

S.No	Expected Error	Reason
1	S	Capital S is expected for String and System.

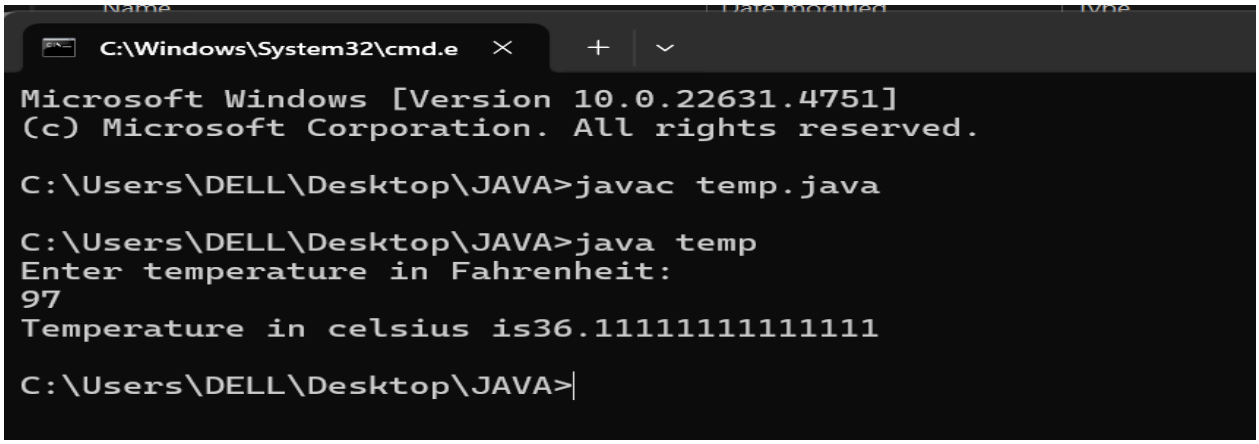
2a). Write a program to convert temperature from Fahrenheit to Celsius.

Code:

```
import java.util.Scanner;
class temp{
        public static void main(String[] args){
                        Scanner scan=new Scanner(System.in);
                        System.out.println("Enter temperature in Fahrenheit:");
                        double f=scan.nextDouble();
                        double c=((f-32)/(1.8));
                        System.out.println("Temperature in celsius is"+c);
        }
}
```

Output:

```
C:\Users\DELL\Desktop\JAVA>javac temp.java

C:\Users\DELL\Desktop\JAVA>java temp
Enter temperature in Fahrenheit:
hi
Exception in thread "main" java.util.InputMismatchException
        at java.base/java.util.Scanner.throwFor(Scanner.java:947)
        at java.base/java.util.Scanner.next(Scanner.java:1602)
        at java.base/java.util.Scanner.nextDouble(Scanner.java:2573)
        at temp.main(temp.java:6)

C:\Users\DELL\Desktop\JAVA>
```



Error:

S.No	Expected Error	Reason
1	S	Capital S is expected for String and System.

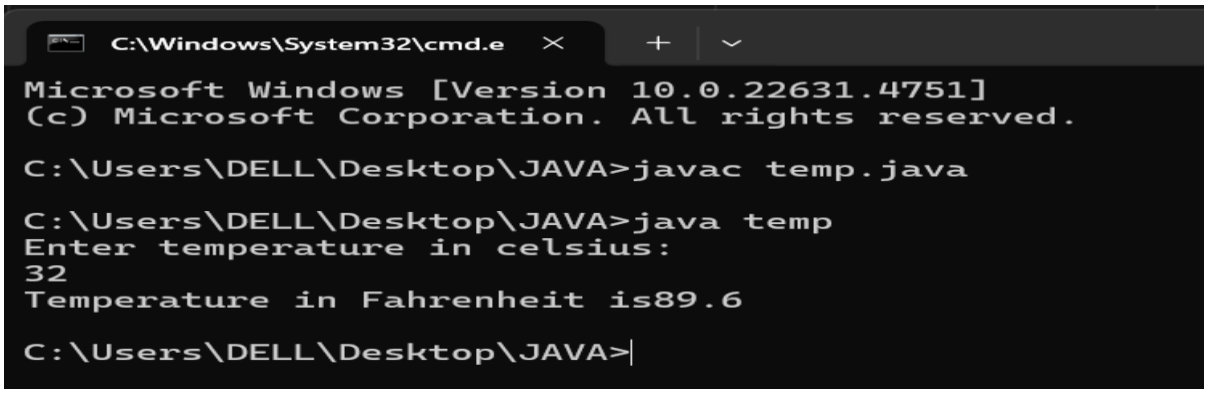
2b). Write a program to convert temperature from Celsius to Fahrenheit.

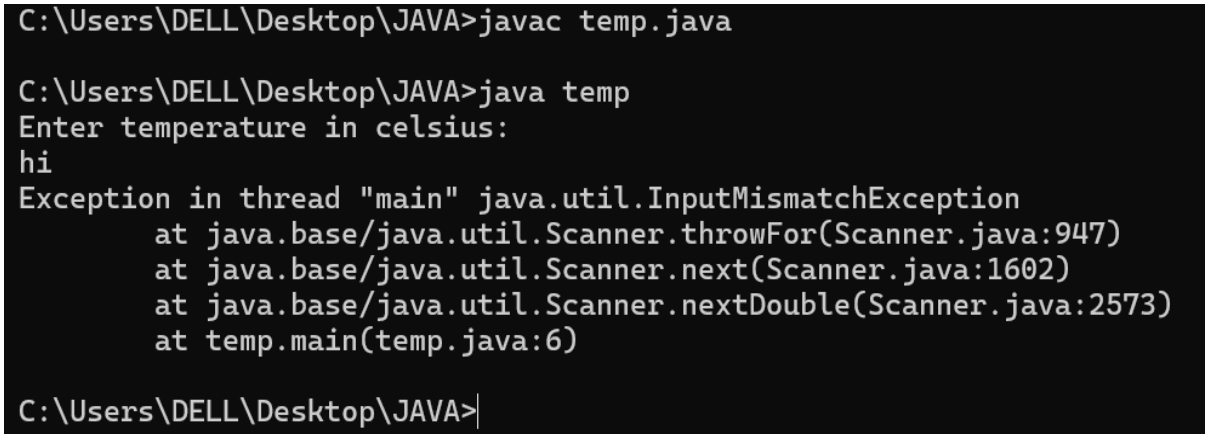
Code:

```
import java.util.Scanner;
class temp{
        public static void main(String[] args){
                        Scanner scan=new Scanner(System.in);
```

```
        System.out.println("Enter temperature in celsius:");
        double c=scan.nextDouble();
        double f=(c*1.8)+32;
        System.out.println("Temperature in Fahrenheit is"+f);
    }
}
```

Output:





ERRORS :

S.No	Expected Error	Reason
1	;	; is expected at end
2	Input.close();	The input is expected to be closed.

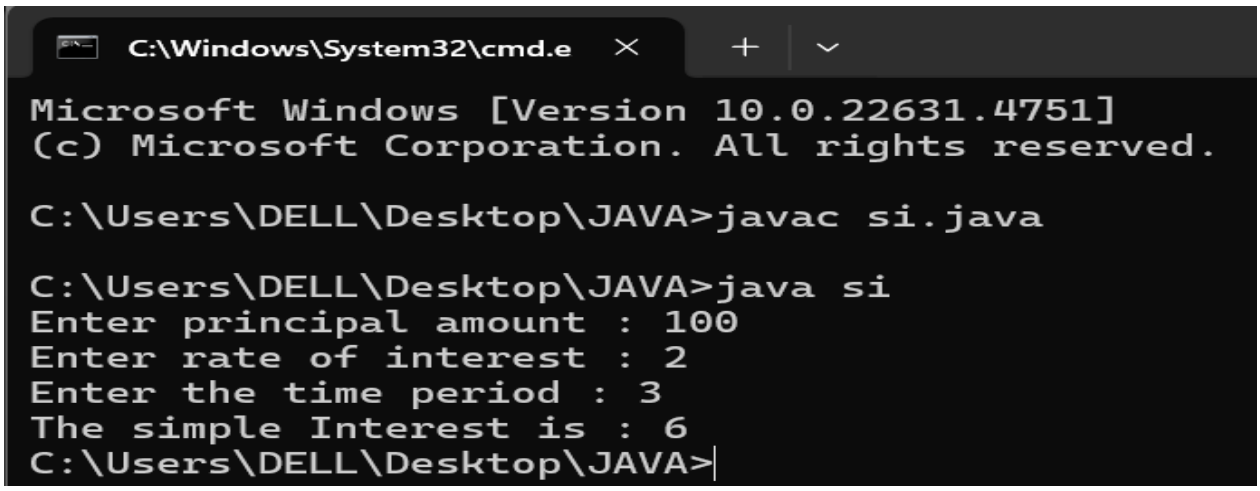
3)Write a java program to calculate the simple interest.

Code:

```
import java.util.Scanner;
        public class si{
         public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        System.out.print("Enter principal amount : ");
        int p = input.nextInt();
        System.out.print("Enter rate of interest : ");
        int r = input.nextInt();
       System.out.print("Enter the time period : ");
     int t = input.nextInt();
     int SI = p*r*t/100;
    System.out.print("The simple Interest is : " + SI);
    input.close();
  }
}
```

Output:

```
C:\Users\DELL\Desktop\JAVA>javac si.java

C:\Users\DELL\Desktop\JAVA>java si
Enter principal amount : h
Exception in thread "main" java.util.InputMismatchException
        at java.base/java.util.Scanner.throwFor(Scanner.java:947)
        at java.base/java.util.Scanner.next(Scanner.java:1602)
        at java.base/java.util.Scanner.nextInt(Scanner.java:2267)
        at java.base/java.util.Scanner.nextInt(Scanner.java:2221)
        at si.main(si.java:6)

C:\Users\DELL\Desktop\JAVA>
```



ERRORS :

S.No	Expected Error	Reason
1	;	; is expected at end
2	Int t	Without declaring t the compiler cannot execute the program.

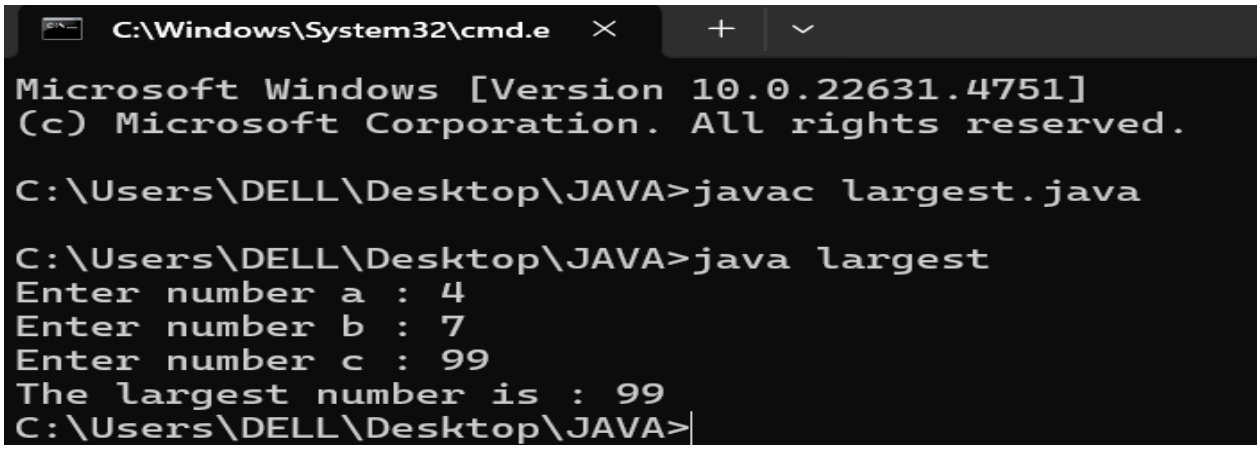
4) Write a java program to find the largest of three numbers using ternary operation.

Code:

```
import java.util.Scanner;
        public class largest{
         public static void main(String[] args){
         Scanner input = new Scanner(System.in);
         System.out.print("Enter number a : ");
         int a = input.nextInt();
         System.out.print("Enter number b : ");
         int b = input.nextInt();
        System.out.print("Enter number c : ");
       int c = input.nextInt();
```

```
        int largest = (a>=b) ? ((a>=c ) ? a : c) : ((b >=c) ? b : c);
        System.out.print("The largest number is : " + largest);
        input.close();
    }
}
```

Output:



ERRORS :

S.No	Expected Error	Reason
1	?	Checks the condition
2	:	Comparing between two variables

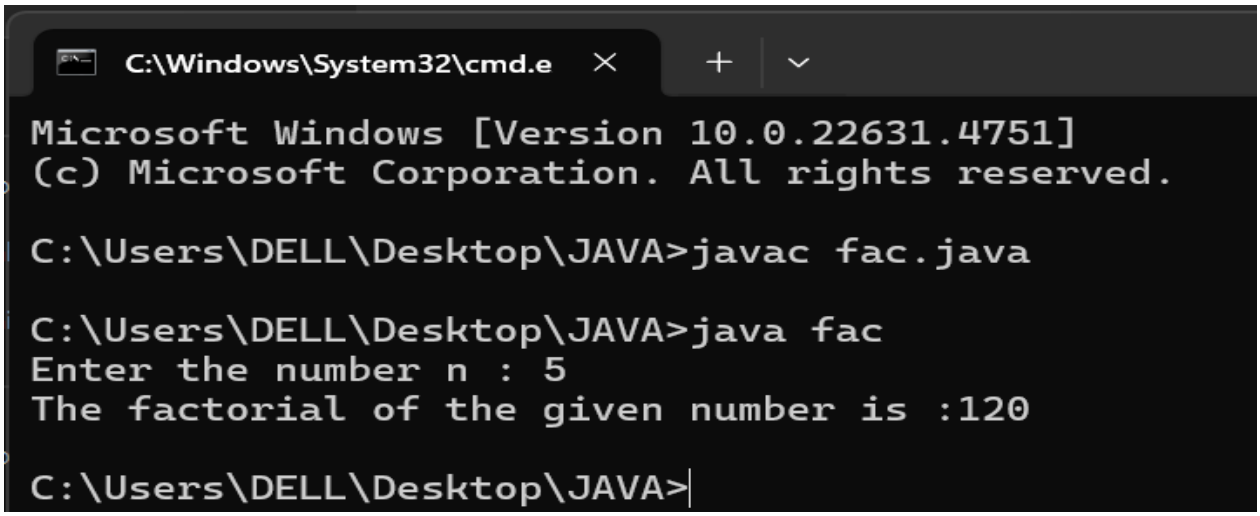
5) Write a java program to find the factorial of a number

Code:

```
import java.util.Scanner;
    public class fac{
        public static void main(String[] args){
      Scanner input = new Scanner(System.in);
      System.out.print("Enter the number n : ");
```

```
        int n = input.nextInt();
        int fac = 1;
            for(int i = 2; i<=n;i++){
        fac *= i;
        }
    System.out.println( "The factorial of the given number is :" + fac);
    input.close();
  }
}
```

Output:



ERRORS :

S.No	Expected Error	Reason
1	}	To close for loop
2	System.out.print();	If we place the print statement inside the for loop it will print the each i value everytime but to print only the final value we must place it outside the for loop.

WEEK – 3

1)Create the java program with the following instructions

i) Create a class with name Car
ii) Create 4 attributes named Car_Color , Car_brand, fuel_type, mileage
iii) Create 3 method named Start( ) , Stop( ), Service( )
iv) Create 3 objects Car1 , Car2 , Car3
v) Create a constructor which should print “Welcome to Car Garage”

Code:

```
public 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","Audi","Petrol",20);
Car Car2 = new Car("White","BMW","Diesel",17);
Car Car3 = new Car("Red","Mercedes","Petrol",16);
Car1.Start();
}
}
```

Output:

```
C:\Users\DELL\Desktop\JAVA>javac Car.java

C:\Users\DELL\Desktop\JAVA>java Car
Black Audi Petrol 20
White BMW Diesel 17
Red Mercedes Petrol 16
The car has just started
```

Errors :

S.No	Expected Error	Reason
1	}	} is expected at end of the class
2	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first

Class Diagram:

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

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

i) In deposit( ) whenever an amount is deposited it has to be updated with current amount

ii) In withdraw( ) whenever an amount is withdrawn it has to be less than current amount else print “Insufficient funds”.

Code:

```
public class BankAccount {
   private String Name;
   private int AccNo, CurrBal;

   public BankAccount(String Name, int AccNo, int CurrBal) {
      this.Name = Name;
      this.AccNo = AccNo;
      this.CurrBal = CurrBal;
      System.out.println("The customer is: " + this.Name);
   }

   public int deposit(int dAmt) {
```

```
        CurrBal += dAmt;
        return CurrBal;
    }

    public void withdraw(int wAmount) {
        if (wAmount <= CurrBal) {  // Allowing withdrawal if balance is equal
            CurrBal -= wAmount;
            System.out.println("Remaining Balance: " + CurrBal);
        } else {
            System.out.println("Insufficient funds");
        }
    }

    public static void main(String[] args) {
        BankAccount Vanshika = new BankAccount("Vanshika", 1500, 10000);

        Vanshika.withdraw(13000);  // Should print "Insufficient funds"
        Vanshika.withdraw(1900);   // Should print remaining balance
        int FinalAmount = Vanshika.deposit(10000);
        System.out.println("Final Balance: " + FinalAmount);
    }
}
```

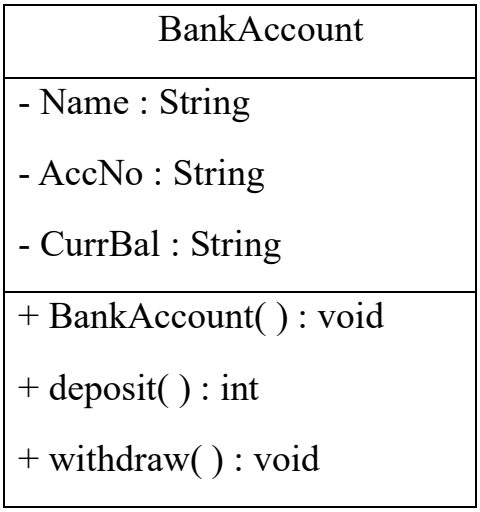
Output:

```
C:\Users\DELL\Desktop\JAVA>javac BankAccount.java

C:\Users\DELL\Desktop\JAVA>java BankAccount
The customer is: Vanshika
Insufficient funds
Remaining Balance: 8100
Final Balance: 18100
```

Errors:

S.No	Expected Error	Reason
1	}	} is expected at end of the class
2	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first

Class Diagram:



WEEK – 4

1) 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.

Code:

```
import java.util.Scanner;
class book{
    public String title;
    public String author;
    public int year;
    book(String title,String author, int year){
            this.title=title;
            this.author=author;
            this.year=year;
    }
    public void display(){
            System.out.println("Title of the book is: "+title);
            System.out.println("Author of the book is: "+author);
            System.out.println("Year of publishion of the book is: "+year);
    }
    public static void main(String[] args){
            Scanner scan=new Scanner(System.in);
            System.out.println("Enter name of the book:");
```

```
        String title=scan.nextLine();
        System.out.println("Enter author of the book:");
        String author=scan.nextLine();
        System.out.println("Enter year of publishion of the book:");
        int year=scan.nextInt();
        book third=new book(title,author,year);
        third.display();
        book first=new book("The kill a mocking bird","Harper Lee",2005);
        book second=new book("The alchemist","Paulo Coelho",1995);
        first.display();
        second.display();
    }
}
```

Output:

```
C:\Users\DELL\Desktop\JAVA>javac book.java

C:\Users\DELL\Desktop\JAVA>java book
Enter name of the book:
The monk who sold his ferrari
Enter author of the book:
Robin Sharma
Enter year of publishion of the book:
1996
Title of the book is: The monk who sold his ferrari
Author of the book is: Robin Sharma
Year of publishion of the book is: 1996
Title of the book is: The kil a mocking bird
Author of the book is: Harper Lee
Year of publishion of the book is: 2005
Title of the book is: The alchemist
Author of the book is: Paulo Coelho
Year of publishion of the book is: 1995
```

Errors :

S.No.	Expected Error	Reason
1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required

Class Diagram:

book
+ title : String + author : String + year : int
+ display() : void

2) 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;
        myclass(){
                count++;
        }
        void display(){
        System.out.println("The value of pi is: "+ pi);
        }
        public static void main(String[] args){
        myclass obj1=new myclass();
        myclass obj2=new myclass();
        myclass obj3=new myclass();
        int fc=count;
        System.out.println("Total number of objects created is: "+fc);
        obj1.display();
        obj2.display();
        obj3.display();
        }
}
```

Output:

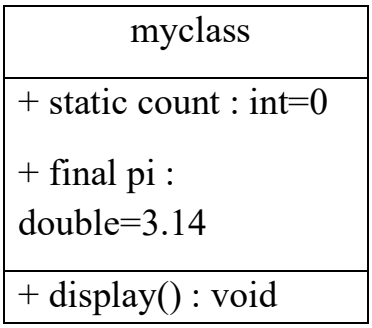
```
C:\Users\DELL\Desktop\JAVA>javac myclass.java

C:\Users\DELL\Desktop\JAVA>java myclass
Total number of objects created is: 3
The value of pi is: 3.1415
The value of pi is: 3.1415
The value of pi is: 3.1415
```

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

Class Diagram:



WEEK –5

1) Create a calc using the operations including add, sub, mul, div using multilevel inheritance and display the desired output.

Code:

```
class bcalc {
   int a, b;
   int sum, diff;
   bcalc(int a, int b) {
      this.a = a;
      this.b = b;
   }
   public void add() {
      diff = a - b;
      sum = a + b;
      System.out.println("Difference: " + diff);
      System.out.println("Sum: " + sum);
   }
}
class acalc extends bcalc {
   int mul;
   acalc(int a, int b) {
      super(a, b);
   }
   public void mult() {
      mul = a * b;
      System.out.println("Multiplication: " + mul);
```

```
    }
}
class aacalc extends acalc {
    float div;
    aacalc(int a, int b) {
        super(a, b);
    }
    public void divi() {
        if (b != 0) {  // Check to avoid division by zero
            div = (float) a / b;
            System.out.println("Division: " + div);
        } else {
            System.out.println("Division by zero error!");
        }
    }
}
class ocalc {
    public static void main(String[] args) {
        aacalc c = new aacalc(10, 2);
        c.divi();
        c.mult();
        c.add();
    }
}
```

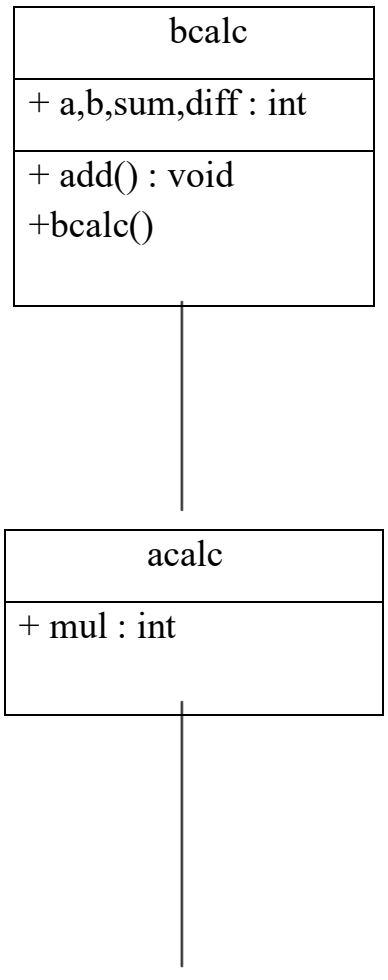
Output:

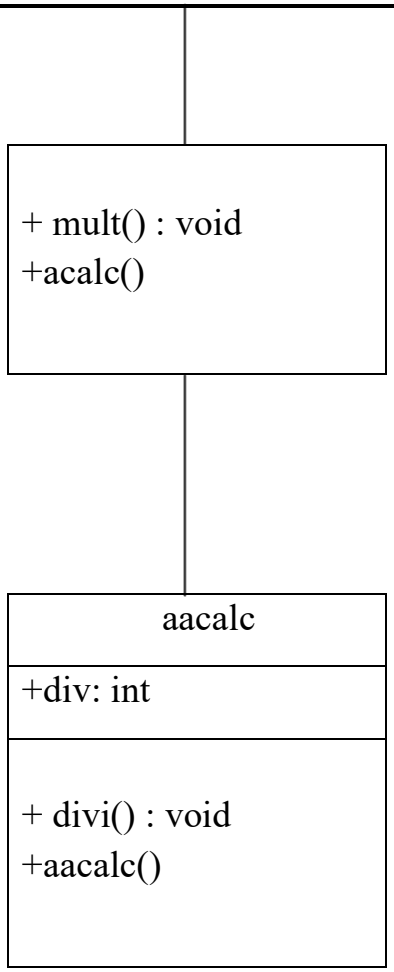
```
C:\Users\DELL\Desktop\JAVA>javac ocalc.java

C:\Users\DELL\Desktop\JAVA>java ocalc
Division: 5.0
Multiplication: 20
Difference: 8
Sum: 12
```

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

Class Diagram:




+ mult() : void
+acalc()
aacalc
+div: int
+ divi() : void
+aacalc()

2) A vehicle rental company wants to develop a system that maintains information about different types

of vehicles available for rent. The company rents out cars and bikes and they need a program to store

details about each vehicle such as brand and speed

cars should have an additional properties(attributes)- no.of doors , seating capacity

bikes should have a property indicating whether they have gears or not

the system should also include a function to display details about each vehicle and indicate when a vehicle is starting

each class should have a constructor .

a) which oops concept is used in the above program ? Explain why it is useful in this scenario

b)If the company decides to add a new type of vehicle truck how would u modify the above program

1) truck should include an additional property called capacity(in tons)

2)create a show truck details method() to display the trucks capacity

3)write a constructor for truck that initializes all the properties

c)Implement the truck class and update the main method to create the truck object and also create an object for car and bike subclass. Finally display its details.

Code:

```
// OOP Concept: Inheritance and Polymorphism are used in this program.

// Inheritance allows different vehicle types to share common properties from a base class.

// Polymorphism enables a generic function to display vehicle details dynamically.


// Base class

class Vehicle {
```

```
    String brand;
    int speed;

    Vehicle(String brand, int speed) {
        this.brand = brand;
        this.speed = speed;
    }

    void displayDetails() {
        System.out.println("Brand: " + brand);
        System.out.println("Speed: " + speed + " km/h");
    }

    void startVehicle() {
        System.out.println(brand + " is starting...");
    }
}

// Car subclass
class Car extends Vehicle {
    int noOfDoors;
    int seatingCapacity;

    Car(String brand, int speed, int noOfDoors, int seatingCapacity) {
        super(brand, speed);
        this.noOfDoors = noOfDoors;
        this.seatingCapacity = seatingCapacity;
```

```
    }

    @Override
    void displayDetails() {
        super.displayDetails();
        System.out.println("Number of Doors: " + noOfDoors);
        System.out.println("Seating Capacity: " + seatingCapacity);
    }
}

// Bike subclass
class Bike extends Vehicle {
    boolean hasGears;

    Bike(String brand, int speed, boolean hasGears) {
        super(brand, speed);
        this.hasGears = hasGears;
    }

    @Override
    void displayDetails() {
        super.displayDetails();
        System.out.println("Has Gears: " + (hasGears ? "Yes" : "No"));
    }
}

// Truck subclass
```

```
class Truck extends Vehicle {
    double capacity;

    Truck(String brand, int speed, double capacity) {
        super(brand, speed);
        this.capacity = capacity;
    }

    void showTruckDetails() {
        System.out.println("Truck Capacity: " + capacity + " tons");
    }

    @Override
    void displayDetails() {
        super.displayDetails();
        showTruckDetails();
    }
}

// Main class
public class VehicleRentalSystem {
    public static void main(String[] args) {
        Car car = new Car("Toyota", 150, 4, 5);
        Bike bike = new Bike("Yamaha", 120, true);
        Truck truck = new Truck("Volvo", 100, 15.5);

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

```
        car.displayDetails();
        car.startVehicle();
        System.out.println();

        System.out.println("Bike Details:");
        bike.displayDetails();
        bike.startVehicle();
        System.out.println();

        System.out.println("Truck Details:");
        truck.displayDetails();
        truck.startVehicle();
    }}
```

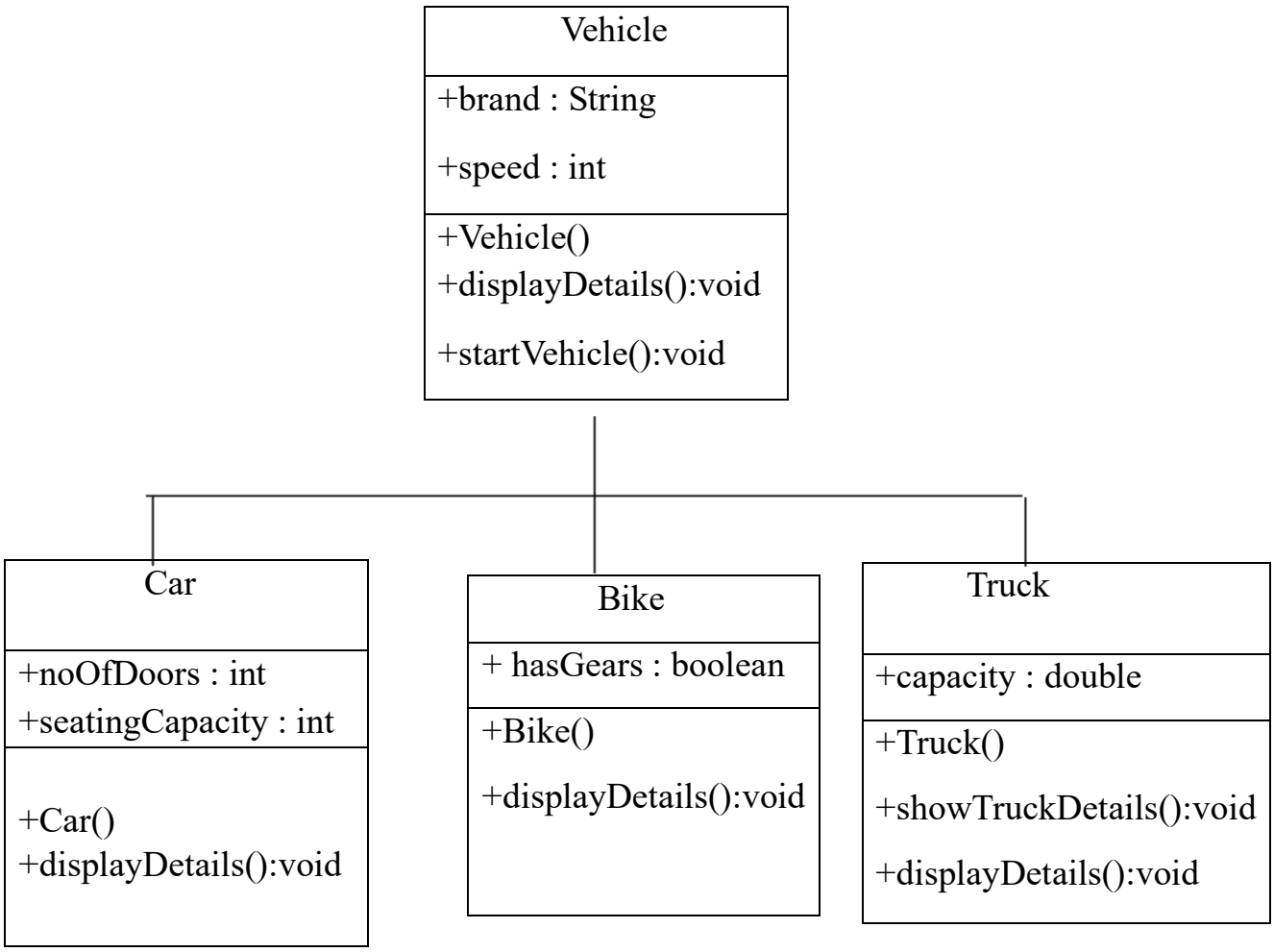
Output:

```
C:\Users\DELL\Desktop\JAVA>java VehicleRentalSystem
Car Details:
Brand: Toyota
Speed: 150 km/h
Number of Doors: 4
Seating Capacity: 5
Toyota is starting...

Bike Details:
Brand: Yamaha
Speed: 120 km/h
Has Gears: Yes
Yamaha is starting...

Truck Details:
Brand: Volvo
Speed: 100 km/h
Truck Capacity: 15.5 tons
Volvo is starting...
```

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

Class Diagram:


Vehicle
+brand : String
+speed : int
+Vehicle()
+displayDetails():void
+startVehicle():void
Car
+noOfDoors : int
+seatingCapacity : int
+Car()
+displayDetails():void
Bike
+ hasGears : boolean
+Bike()
+displayDetails():void
Truck
+capacity : double
+Truck()
+showTruckDetails():void
+displayDetails():void

WEEK-6

1. Write a java program to create a Vehicle class with displayInfo() method , overridden in Car subclass to provide info about carcompany , model , price, seating and petrol.

CODE:

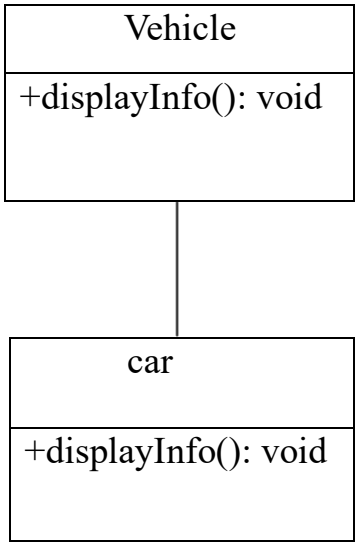
```
class Vehicle{
    public void displayInfo(String comp,String model,int price,int seating,boolean petrol){

        System.out.println("Details");
    }
}

class car extends Vehicle{
    public void displayInfo(String comp,String model,int price,int seating,boolean petrol){
        System.out.println("Car Details");
        System.out.println("Car company:"+comp);
        System.out.println("Car model:"+model);
        System.out.println("Car seating:"+seating);
        System.out.println("Car price:"+price);
        System.out.println("Petrol:"+petrol);
    }
}
```

```
class maruti{
        public static void main(String[] args){
                car c=new car();
                c.displayInfo("maruti","1",1000000,5,true);
        }
}
```

OUTPUT:

```
Car Details
Car company:maruti
Car model:1
Car seating:5
Car price:1000000
Petrol:true
```

CLASS DIAGRAM:



ERROR:

S.No.	Expected Error	Reason
1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required

2. An automated admission system that verifies student eligibility for UG and PG with different criteria.

.UG requires minimum of 60%

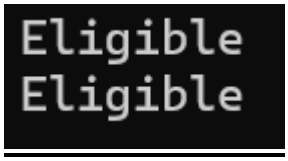
.PG requires minimum of 70%

CODE:

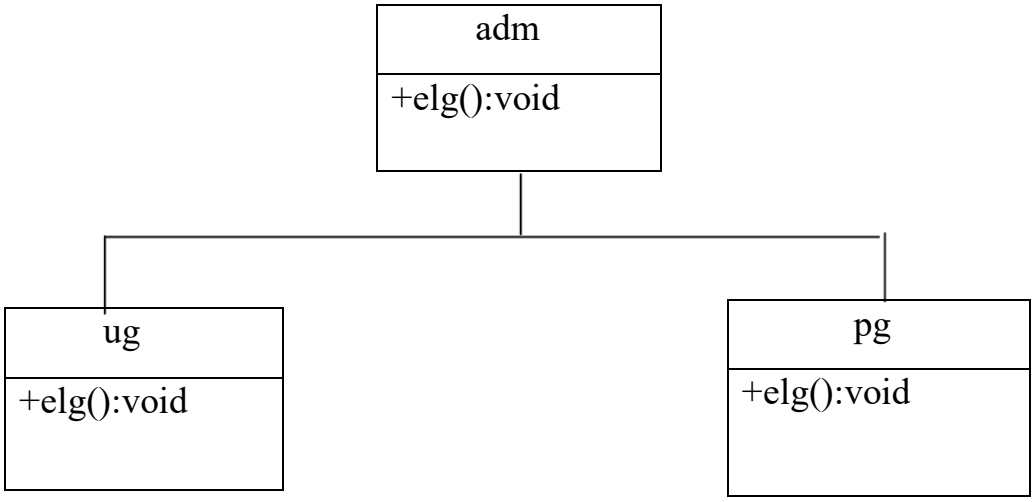
```
class adm{
	public void elg(float score){
		System.out.println("Eligibility");
	}
}
class ug extends adm{
	public void elg(float score){
		if(score>=60){
			System.out.println("Eligible");
		}
```

```
            else{
                System.out.println("Not Eligible");
            }
        }
}
class pg extends adm{
        public void elg(float score){
            if(score>=70){
                System.out.println("Eligible");
            }
            else{
                System.out.println("Not Eligible");
            }
        }
}
class score{
        public static void main(String[] args){
            ug stu1=new ug();
            pg stu2=new pg();
            stu1.elg(85);
            stu2.elg(70);
        }
}
```

OUTPUT:



CLASS DIAGRAM:



ERROR:

S.No.	Expected Error	Reason
1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required

3.Create a calculator class with overloaded methods to perform additions

.add two integers

.add two double values

.add three integers

CODE:

```
class cal{
	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 ocal{
	public static void main(String[] args){
		cal c=new cal();
		System.out.println(c.add(2,3));
		System.out.println(c.add(2.5,3.5));
		System.out.println(c.add(2,3,4));
	}
}
```

OUTPUT:

CLASS DIAGRAM:

cal
+add(int a,int b):int +add(double a,double b):double +add(int a,int b,int c):int

ERROR:

S.No.	Expected Error	Reason
1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required

4. Create a shape class with method calculateArea() that is overloaded for different shapes (eg: square, rectangle).Then create a subclass Circle that overrides calculateArea() method for Circle.

CODE:

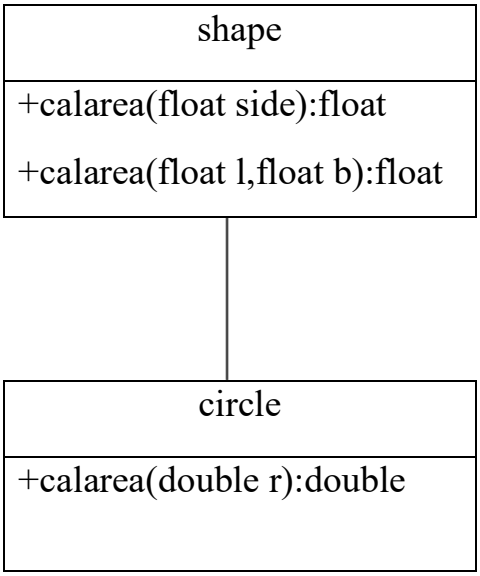
```
class shape{
    public float calarea(float side){
        return side*side;
    }
```

```
        public float calarea(float l,float b){
                return l*b;
        }
}
class circle extends shape{
        public double calarea(double r){
                return 3.14*r*r;
        }
}
 class s{
        public static void main(String[] args){
                circle c=new circle();
                System.out.println(c.calarea(4));
        }
}
```

OUTPUT:

```
16.0
```

CLASS DIAGRAM:



ERROR:

S.No.	Expected Error	Reason
1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required

WEEK-7

1) Write a java program to create an abstract class Animal with abstract method sound and create subclasses Lion and Tiger that implements the method.

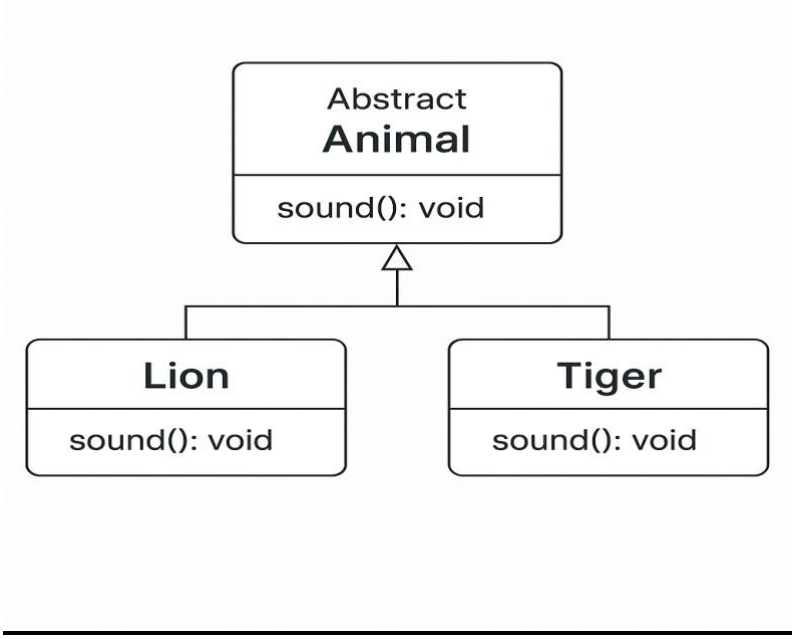
CODE:

```
abstract class Animal{
	public abstract void sound();
}
class Lion extends Animal{
	public void sound(){
		System.out.println("Lion Roars");
	}
}
class Tiger extends Animal{
	public void sound(){
		System.out.println("Tiger Growls");
	}
}
class Animalsound{
	public static void main(String[] args){
		Lion l=new Lion();
		Tiger t=new Tiger();
		l.sound();
		t.sound();
	}}
```

OUTPUT:

```
Lion Roars
Tiger Growls
```

CLASS DIAGRAM:



ERROR:

S.No.	Expected Error	Reason
1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required

2) Write a java program to create an abstract class shape3D with abstract methods to calculate volume and surfacearea and create subclasses for sphere and cube that implements these methods.

CODE:

```
abstract class Shape3D{
        double a;
        Shape3D(double a){
                this.a=a;
        }
        abstract void vol();
        abstract void surfarea();
}
class Sphere extends Shape3D {

   Sphere(double r) {
      super(r);
   }

   void vol() {
      double vol = (4.0 / 3) * Math.PI * a * a * a;
      System.out.println("Volume of sphere with radius " + a + " is " + vol);
   }

   void surfarea() {
      double surfArea = 4 * Math.PI * a * a;
```

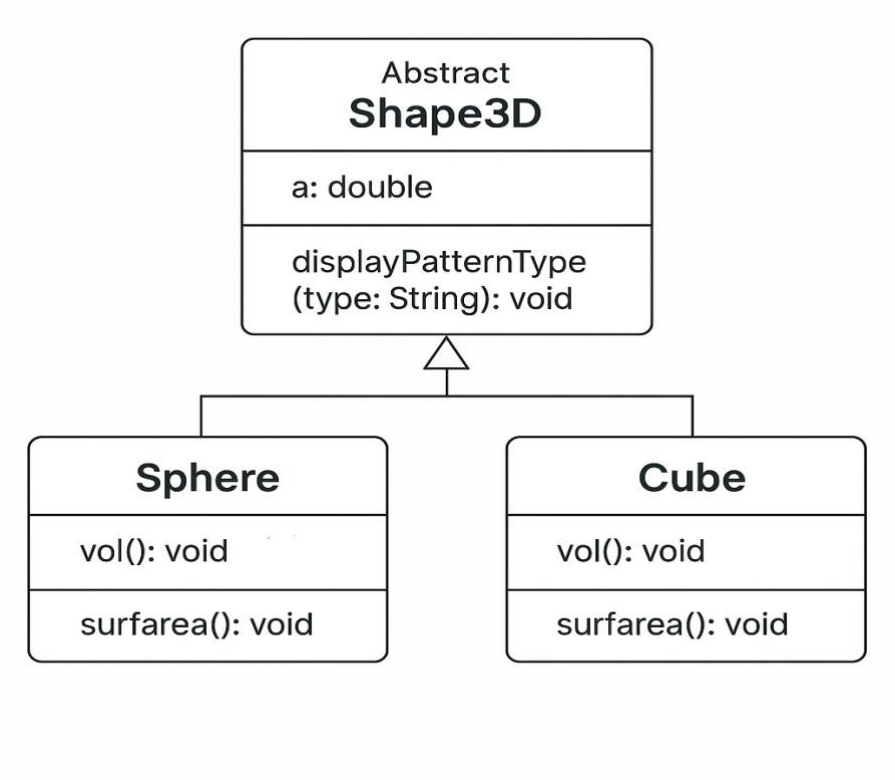
```
        System.out.println("Surface Area of sphere with radius " + a+ " is " +
surfArea);
    }
}
class Cube extends Shape3D {

    Cube(double side) {
        super(side);
    }

    void vol() {
        double vol = a*a*a;
        System.out.println("Volume of cube with side " + a + " is " + vol);
    }

    void surfarea() {
        double surfArea = 6 *  a * a;
        System.out.println("Surface Area of cube with side " + a+ " is " + surfArea);
    }
}
class ssc{
        public static void main(String[] args){
                Cube c=new Cube(2);
                Sphere s=new Sphere(3);
                c.vol();
                c.surfarea();
                s.vol();
```

```
            s.surfarea();
        }
}
```

OUTPUT:

```
Volume of cube with side 2.0 is 8.0
Surface Area of cube with side 2.0 is 24.0
Volume of sphere with radius 3.0 is 113.09733552923255
Surface Area of sphere with radius 3.0 is 113.09733552923255
```

CLASS DIAGRAM:



ERROR:

S.No.	Expected Error	Reason
1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required

3) Create an abstract class PatternPrint with an abstract method printing to print the pattern and a concrete method to display the pattern . Implement the patterns

1) Star Pattern - prints a right angled triangle of stars

2) Number Pattern – prints a right angled triangle of increasing numbers.

CODE:

```
abstract class PatternPrint  {
   abstract void printing();
   void displayPatternType(String type) {
      System.out.println("Pattern Type: " + type);
   }
}

class StarPattern extends PatternPrint {

   void pattern() {
      System.out.println("Generating Star Pattern:");
   }
```

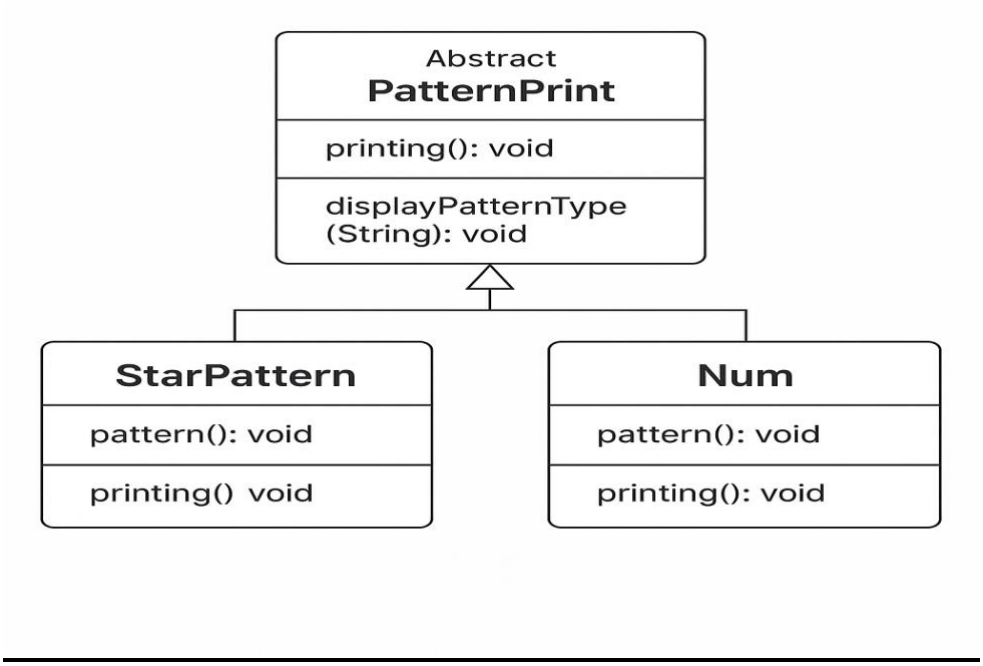
```
    void printing() {
        for (int i = 1; i <= 5; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}

class num extends PatternPrint{
        void pattern() {
        System.out.println("Generating Numbers Pattern:");
    }

    void printing() {
         int k=1;
        for (int i = 1; i <= 5; i++) {
            for (int j = 1; j <= i; j++) {

                    System.out.print(k+"  ");
                            k++;
                    }

            System.out.println();
        }
    }
```

```
}
public class PatternDemo {
    public static void main(String[] args) {
        StarPattern sp = new StarPattern();
        sp.displayPatternType("Star Triangle");
        sp.pattern();
        sp.printing();
          num n = new num();
        n.displayPatternType("Increasing Numer Triangle");
        n.pattern();
        n.printing();


    }
}
```

OUTPUT:

```
Pattern Type: Star Triangle
Generating Star Pattern:
*
* *
* * *
* * * *
* * * * *
Pattern Type: Increasing Numer Triangle
Generating Numbers Pattern:
1
2  3
4  5  6
7  8  9  10
11  12  13  14  15
```

CLASS DIAGRAM:



ERROR:

S.No.	Expected Error	Reason
1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required