

23CSE111

OBJECT ORIENTED PROGRAMMING

LAB REPORT



Department of Computer Science Engineering

Amrita School of Computing

Amrita Vishwa Vidhyapeetam, Amaravati Campus

Name:M.AKSHAYA

Verified By

Roll No: 24218

WEEK-1

1) Explain the process of Installing JDK (Java Development Kit)

Installing of JDK (Java Development Kit):

1. Download JDK:

- Go to the Oracle JDK download page in your web browser and click on JDK-21 version which is Long term support (LTS) version.
- Click on the download link for your operating system (Windows, macOS, or Linux).

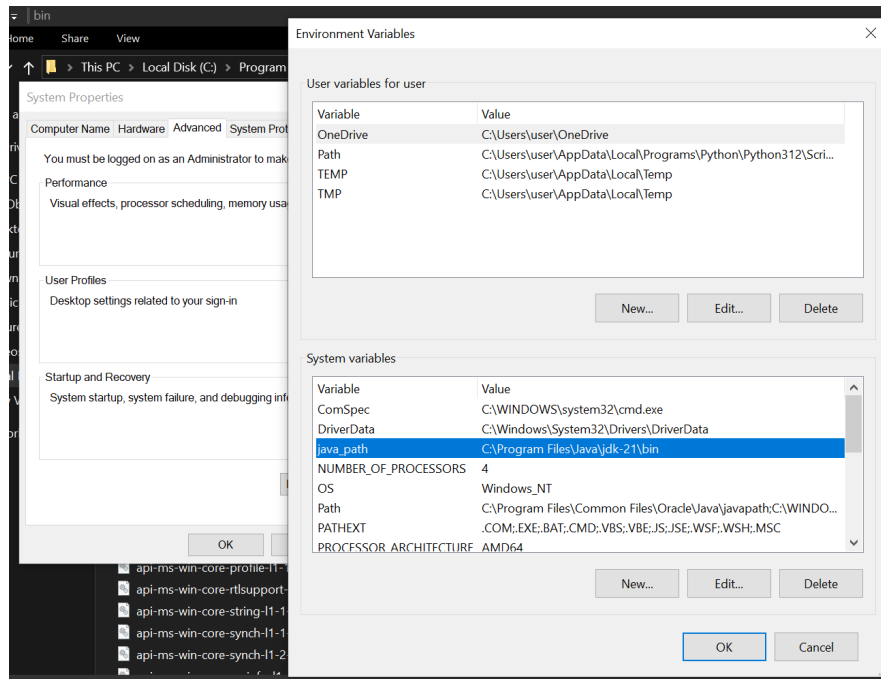
2. Install JDK:

- Once downloaded, run the installer.
- Follow the instructions and keep clicking "Next" until it's done.

3. Set Environment Variables (Windows):

- Open file explorer, then right click on This PC next select on properties then it will take you to the settings app then click on advanced system settings and then click on **Environment Variables**.
- Click **New** under **System Variables**:
 - **Set Variable name as:** java_home
 - **Variable value:** The folder address where JDK is installed (like C:\Program Files\Java\jdk-21\bin)
- Find Path under **System Variables**, click **Edit**, and add the path of the jdk-21(C:\Program Files\Java\jdk-21\bin)

○



Checking of JDK Version:

1. **Open Command Prompt:**
 - Press win+R, type cmd, and press Enter.
2. **Check Version:**
 - Type java --version and press Enter.
 - Type javac --version and press Enter.

```

C:\Users\user>javac --version
javac 21.0.5

C:\Users\user>java --version
java 21.0.5 2024-10-15 LTS
Java(TM) SE Runtime Environment (build 21.0.5+9-LTS-239)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.5+9-LTS-239, mixed mode, sharing)

C:\Users\user>

```

Program-1:

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

Print helloworld:-

```
public class helloworld {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

Output:

```
m.akshaya@Akshaya-MacBook-Air MyProjects % javac Helloworld.java  
m.akshaya@Akshaya-MacBook-Air MyProjects % java Helloworld  
Hello World  
m.akshaya@Akshaya-MacBook-Air MyProjects %
```

ERRORS: None found

Program-2:

AIM: Write a Java Program that prints Name, Roll No, Section of a student.

STUDENT DETAILS:

```
public class studentinformation {  
    public static void main(String [] args) {  
        System.out.println("NAME:M.AKSHAYA");  
        System.out.println("Section:C");  
        System.out.println("Roll no:AV.SC.U4CSE24218");  
    }  
}
```

OUTPUT:

```
m.akshaya@Akshaya-MacBook-Air MyProjects % javac Studentinformation.java  
m.akshaya@Akshaya-MacBook-Air MyProjects % java Studentinformation  
NAME:M.AKSHAYA  
Section:C  
Roll no:AV.SC.U4CSE24218  
m.akshaya@Akshaya-MacBook-Air MyProjects %
```

IMPORTANT POINTS:

1. When printing the statements, everything should be inside double quotes.

ERROR TABLE:

Code Error	Code rectification
1) writing small "S" in place of "S" In system.out.println() 2) not giving strings to the name and section	1) code is rectified by keeping capital "S" 2) Giving strings to name and section

WEEK-2:

Program-1:

AIM: Write a java program to Calculate area of rectangle.

CODE:

```
import java.util.scanner;  
Public class rectangle{  
    Public static void main(String[]args){  
        Scanner scanner=new Scanner(system.in);  
        System.out.println("enter length");  
        Double length=scanner.new double();  
        System.out.println("enter the width")  
        Double width=scanner.new double();  
        Double area=length*width  
        System.out.println("area"+area);  
    }  
}
```

```

        scanner.close();
    }
}

```

OUTPUT:

```

m.akshaya@Akshaya-MacBook-Air MyProjects % javac rectangle.java
m.akshaya@Akshaya-MacBook-Air MyProjects % java recatngle
Error: Could not find or load main class recatngle
Caused by: java.lang.ClassNotFoundException: recatngle
m.akshaya@Akshaya-MacBook-Air MyProjects % java rectangle
Enter the length of the rectangle: 10
Enter the width of the rectangle: 5
The area of the rectangle is: 50.0

```

ERROR TABLE:

Code Error	Code rectification
<ol style="list-style-type: none"> 1. While using for iteration, not giving the conditions correctly. 2. Declaring the data type as double instead of int. 3. Writing small 	<ol style="list-style-type: none"> 1. We should give iterative statements correctly. 2. We should give the data type as int for integers.

IMPORTANT POINTS:

- 1.Area of a rectangle is $\text{area} = l \times b$, where
 L = length of a side of the rectangle,
 B = breadth of a side of the rectangle.
- 2.Here, we must be sure that all the expressions/conditions inside for the for loop must be given correctly

Program-2:

AIM: Write a java program to Calculate the simple interest by input given by the user.

CODE:

```
Import java.util.Scanner;
public class SimpleInterest {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the principal amount (P): ");
        double principal = scanner.nextDouble();
        System.out.print("Enter the rate of interest (R) in
percentage: ");
        double rate = scanner.nextDouble();
        System.out.print("Enter the time period (T) in years: ");
        double time = scanner.nextDouble();
        double simpleInterest = (principal * rate * time) / 100;
        System.out.println("The Simple Interest is: " +
simpleInterest);
        scanner.close();
    }
}
```

OUTPUT:

```
m.akshaya@Akshaya-MacBook-Air MyProjects % javac SimpleInterest.java
m.akshaya@Akshaya-MacBook-Air MyProjects % java SimpleInterest
Enter the principal amount (P): 25000
Enter the rate of interest (R) in percentage: 2
Enter the time period (T) in years: 5
The Simple Interest is: 2500.0
```

ERROR TABLE:

Code Error	Code rectification
------------	--------------------

<ol style="list-style-type: none"> 1. Giving space between next and Double. 2. Not giving parenthesis after closing the input. 	<ol style="list-style-type: none"> 1. Should not give space between next and Double. 2. We must put parenthesis after closing the input.
--	--

IMPORTANT POINTS:

1.Simple interest formula is: $(p*t*r)/100$, where:

P: Principal amount

R: Rate of interest

T: Time period

2.The data type double indicates the floating points in the integers.

3.The line “import java.util.Scanner” indicates:

Import: tells the java compiler that we want to use a specific class or package in your code.

Java.util : This is the package that contains utility classes for Java programming, including the “Scanner” class.

Scanner: this is the class that allows you to read input from the keyboard.

Program-3:

AIM: Write a java program to calculate the Factorial of N(given by the user).

CODE:

```
import java.util.Scanner;
public class factorial {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number: ");
        int number = scanner.nextInt();
        long factorial = 1;
```



```

        if (number < 0) {
            System.out.println("No Factorial For Negative
Numbers.");
        } else {
            for (int i = 1; i <= number; i++) {
                factorial *=i;
            }
            System.out.println("The factorial of " + number + " is " +
factorial);
        }
        scanner.close();
    }
}

```

OUTPUT:

```

m.akshaya@Akshaya-MacBook-Air MyProjects % javac factorial.java
m.akshaya@Akshaya-MacBook-Air MyProjects % java factorial
Enter the number: 6
The factorial of 6 is 720

```

ERROR TABLE:

Code Error	Code rectification
3. While using for iteration, not giving the conditions correctly. 4. Declaring the data type as double instead of int.	3. We should give iterative statements correctly. 4. We should give the data type as int for integers.

IMPORTANT POINTS:

- 1.While the for loop the data inside the parenthesis indicates the Initial expression
Test expression and
Update expression.
- 2.Here “factorial*=i” means factorial = factorial*i.

3. Here we are using the data type "int" just to calculate the integer values and it doesn't support floating points.

Program-4:

AIM: Write a java program to find the Fibonacci series (all inputs taken from the user).

CODE:

```
public class FibonacciSeries {  
    public static void main(String[] args) {  
        int n = 10;  
        int firstTerm = 0, secondTerm = 1;  
        System.out.println("Fibonacci Series up to " + n + "  
terms:");  
        for (int i = 1; i <= n; ++i) {  
            System.out.print(firstTerm + ", ");  
            int nextTerm = firstTerm + secondTerm;  
            firstTerm = secondTerm;  
            secondTerm = nextTerm;  
        }  
    }  
}
```

OUTPUT:

```
m.akshaya@Akshaya-MacBook-Air MyProjects % touch fibonacci.java  
m.akshaya@Akshaya-MacBook-Air MyProjects % javac fibonacci.java  
m.akshaya@Akshaya-MacBook-Air MyProjects % java fibonacci  
Fibonacci Series up to 10 terms:  
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, %
```

ERROR TABLE:

Code Error	Code rectification
------------	--------------------

<ol style="list-style-type: none"> 1. Giving space between next and Double. 2. Not giving parenthesis after closing the input. 	<ol style="list-style-type: none"> 1. Should not give space between next and Double. 2. We must put parenthesis after closing the input.
--	--

IMPORTANT POINTS:

1. In the Fibonacci sequence, the sum value is given to the second variable, and the value of the second variable is given to the first variable.
2. This process is repeated a certain number of times until the conditions are met.

Program-5:

AIM: Write a java program to find the area of triangle using herons formula.

CODE:

```
import java.util.Scanner;
public class triangle{
    public static void main(String[] args){
        Scanner scanner=new Scanner(System.in);
        System.out.println("enter length a:");
        Double a = scanner.nextDouble();
        System.out.println("enter length b:");
        Double b = scanner.nextDouble();
        System.out.println("enter length c:");
        Double c = scanner.nextDouble();
        Double s= (a+b+c/2);
        Double area= Math.sqrt(s*(s-a)*(s-b)*(s-c));
        System.out.println("area of the triangle by heron formula"+
area);
        scanner.close();
    }
}
```

```
}  
}
```

OUTPUT:

```
m.akshaya@Akshaya-MacBook-Air MyProjects % javac triangle.java  
m.akshaya@Akshaya-MacBook-Air MyProjects % java triangle  
enter length a:  
2  
enter length b:  
4  
enter length c:  
6  
area of the triangle by heron formula30.740852297878796
```

ERROR TABLE:

Code Error	Code rectification
<ol style="list-style-type: none">1. While printing the variable not giving + sign.2. Not closing the scanner.	<ol style="list-style-type: none">1. We should give correct indentation.2. Closing the scanner is must.

IMPORTANT POINTS:

1. Here, we're finding the area of a triangle using heron's formula.

2. Heron's formula for finding a triangle is:

$$S = (a + b + c) / 2$$

Where S is the semi-perimeter of the triangle.

Now the area formula is:

$$\text{Area} = \sqrt{s(s-a)(s-b)(s-c)}.$$

Program-6:

I) AIM: Write a java program to Convert temperature celsius into fahrenheit.

CODE:

```
import java.util.Scanner;

public class celsiustofahrenheit {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter temperature in Celsius: ");
        float celsius = scanner.nextFloat();
        float fahrenheit = (celsius * 9 / 5) + 32;
        System.out.println(celsius + "°C is equal to " + fahrenheit
+ "°F");
        scanner.close();
    }
}
```

OUTPUT:

```
m.akshaya@Akshaya-MacBook-Air MyProjects % javac celsiustofahrenheit.java
m.akshaya@Akshaya-MacBook-Air MyProjects % java celsiustofahrenheit
Enter temperature in Celsius: 49
49.0°C is equal to 120.2°F
```

ERROR TABLE:

Code Error	Code rectification
<ol style="list-style-type: none">1. While printing the variable not giving + sign.2. Not closing the scanner.	<ol style="list-style-type: none">1. We should give correct indentation.2. Closing the scanner is must.

II) AIM: Write a java program to Convert temperature Fahrenheit to celsius

CODE:

```
import java.util.Scanner;

public class fahrenheittocelsius {
    public static void main(String[] args) {
        int C, F;
        Scanner num = new Scanner(System.in);

        System.out.println("Enter the Fahrenheit temperature: ");
        F = num.nextInt();

        C = (F - 32) * 5 / 9;
        System.out.println("Celsius is: " + C);

        num.close(); // Close the Scanner to avoid resource leak
    }
}
```

OUTPUT:

```
m.akshaya@Akshaya-MacBook-Air ~ % javac fahrenheittocelsius.java
m.akshaya@Akshaya-MacBook-Air ~ % java fahrenheittocelsius
Enter the Fahrenheit temperature:
54
Celsius is: 12
```

ERROR TABLE:

Code Error	Code rectification
<ol style="list-style-type: none">1. While printing the variable not giving + sign.2. Not closing the scanner.	<ol style="list-style-type: none">1. We should give correct indentation.2. Closing the scanner is must.

IMPORTANT POINTS:

- 1.The formula to convert a Fahrenheit to Celsius is
$$\text{Celsius} = (\text{Fahrenheit} - 32) * 5/9$$
- 2.The formula to convert a Celsius to Fahrenheit is
$$\text{Fahrenheit} = (\text{Celsius} * 9/5) + 32.$$
- 3.The line “Scanner input = new Scanner(System.in),” tends to create a new Scanner object named “input” that reads input from the standard input stream (System.in), like keyboard.

WEEK -3:

1.AIM: To create java program with following instructions :

- 1.Create a class with name Car
- 2.Create four attributes named car_color,car_brand, fuel_type, mileage
- 3.Create these methods named start(),stop(),service()
- 4.Create the objects named car, car1,car2

CODE:

```
public class Car {  
    private String car_color;  
    private String car_brand;  
    private String fuel_type;  
    private String mileage;  
  
    public void start() {  
        System.out.println("car is started");  
    }  
}
```

```
public void stop() {
    System.out.println("car is stopped");
}

public void service() {
    System.out.println("car is for service");
}

public static void main(String args[]) {
    Car car = new Car();
    car.car_color = "white";
    car.car_brand = "audi";
    car.fuel_type = "petrol";
    car.mileage = "20";
    car.start();
    System.out.println("car_color: " + car.car_color + "
car_brand: " + car.car_brand + " fuel_type: " + car.fuel_type + "
mileage: " + car.mileage);
    Car car1 = new Car();
    car1.car_color = "white";
    car1.car_brand = "audi";
    car1.fuel_type = "petrol";
    car1.mileage = "20";
    car1.stop();
    System.out.println("car_color: " + car1.car_color + "
car_brand: " + car1.car_brand + " fuel_type: " + car1.fuel_type
+ " mileage: " + car1.mileage);
    Car car2 = new Car();
    car2.car_color = "white";
    car2.car_brand = "audi";
    car2.fuel_type = "petrol";
    car2.mileage = "20";
    car2.service();
}
```



```

        System.out.println("car_color: " + car2.car_color + "
car_brand: " + car2.car_brand + " fuel_type: " + car2.fuel_type
+ " mileage: " + car2.mileage);
    }
}

```

OUTPUT:

```

m.akshaya@Akshaya-MacBook-Air MyProjects % javac car.java
m.akshaya@Akshaya-MacBook-Air MyProjects % java car
car is started
car_color: white car_brand: audi fuel_type: petrol mileage: 20
car is stopped
car_color: white car_brand: audi fuel_type: petrol mileage: 20
car is for service
car_color: white car_brand: audi fuel_type: petrol mileage: 20

```

2.AIM: To create a class BankAccount with methods deposit() and withdraw() . create two subclasses savings account and checking account override the withdraw () method in each subclass to impose different withdrawal limits and fees.

PROGRAM:

```

import java.util.Scanner;

public class BankAccount {
    private double balance;
    public BankAccount(double initialBalance) {
        this.balance = initialBalance;
    }
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
    }
}

```

```
public void withdraw(double amount) {
    if (balance >= amount) {
        balance -= amount;
        System.out.println("Withdrawn: " + amount);
    } else {
        System.out.println("Insufficient balance.");
    }
}

public double getBalance() {
    return balance;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter initial balance: ");
    double initialBalance = scanner.nextDouble();
    BankAccount account = new BankAccount(initialBalance);

    System.out.print("Enter deposit amount: ");
    double depositAmount = scanner.nextDouble();
    account.deposit(depositAmount);

    System.out.print("Enter withdrawal amount: ");
    double withdrawalAmount = scanner.nextDouble();
    account.withdraw(withdrawalAmount);

    System.out.println("Current balance: " +
account.getBalance());

    scanner.close();
}
}
```

OUTPUT:

```
m.akshaya@Akshaya-MacBook-Air MyProjects4 % javac BankAccount.java
m.akshaya@Akshaya-MacBook-Air MyProjects4 % java BankAccount
Enter initial balance: 100000
Enter deposit amount: 50,000
Deposited: 50000.0
Enter withdrawal amount: 90,000
Withdrawn: 90000.0
Current balance: 60000.0
```

ERROR TABLE:

Code Error	Code rectification
<ol style="list-style-type: none">1. Not putting the semi-colon; after calling the function.2. After Withdrawal, deposit not giving the parenthesis ().	<ol style="list-style-type: none">1. Put the semi-colon after the writing the code.2. After every method, put the parenthesis ().

IMPORTANT POINTS:

1. The condition inside the if statement must be correct.
2. It explains that if the withdrawal money is less than the money in the bank account, then we can withdraw the amount.

CLASS DIAGRAM:

Bank Account
-balance:double
+BankAccount(initialBalance:double) +deposit(amount:double):void +withdraw(amount:double):void

WEEK -4

PROGRAM – 1:

AIM: Write a java program with class named “book”, the class should contain various attributes such as title, author, year of publication it should also contain a constructor with parameters which initializes, title, author, and year of publication.

Create a method which displays the details of the book and display the details of two books.

CODE:

```
public class Book{
    String title;
    String author;
    int year_of_publication;
    public Book (String title, String author, int year_of_publication)
    {
        this.title = title;
        this.author = author;
        this.year_of_publication = year_of_publication;
    }
    public void displayDetails() {
        System.out.println(this.title);
        System.out.println(this.author);
        System.out.println(this. year_of_publication);
        System.out.println();
    }
}

public class Main {
    public static void main(String[]args) {
        Book book1 =new Book ( "the first frost"," Sang Yan", 1997);
        Book book2 =new Book ("hidden love", " Sang xi" , 2007);
        System.out.println("Book1 details: ");
```

```

book1.displayDetails();
System.out.println("Book2 details: ");
book2.displayDetails();
}
}

```

OUTPUT:

```

m.akshaya@Akshaya-MacBook-Air ~ % javac Main.java
m.akshaya@Akshaya-MacBook-Air ~ % java Main
Book1 details:
Title: The First Frost
Author: Sang Yan
Year of Publication: 1997

Book2 details:
Title: Hidden Love
Author: Sang Xi
Year of Publication: 2007

```

ERROR TABLE:

Code Error	Code rectification
<ol style="list-style-type: none"> 1. Not defining the function in a file. 2. Two public class files should not be saved in the same file. 	<ol style="list-style-type: none"> 1. To call the method we must define a function in a file. 2. Two public class files should be saved in different files.

NEGATIVE CASE:

```

m.akshaya@Akshaya-MacBook-Air ~ % javac Book.java
Book.java:17: error: class Main is public, should be declared in a file named Main.java
public class Main {
      ^
1 error
m.akshaya@Akshaya-MacBook-Air ~ %

```

IMPORTANT POINTS:

1. While defining two classes for a code, we must be sure that we save both the classes in separate files.
2. While defining a method we should also define a function to call that method.

CLASS DIAGRAM:

Book

- Title: String
- Author: String
- Year of Publication: int

- + Book(title:String,
author:String,Year of
publication:int)
- + DisplayDetails():void

PROGRAM – 2:

AIM: Create a java Program with class named myclass with static variable count of int type, initialized to zero and a constant variable “pi” of type double initialized to 3.14 as attributes of the class, ow define a constructor for “myclass” that increments the count variable each time an object of my class is created (count++), finally print the final values of count and pi variables create three objects.

CODE:

```
public class myclass {  
    Static int count = 0;
```

```

    Final double pi=3.14;
public myclass () {
count++;
}
public static void main (String[]args) {
    My class obj1 =new myclass();
    My class obj2 =new myclass();
    My class obj3 =new myclass();

    System.out.println("count:" + count);
    System.out.println("value of pi:" + obj1.pi);
    System.out.println("value of pi:" + obj2.pi);
    System.out.println("value of pi:" + obj3.pi);
    System.out.println("M.Akshaya");
    }
}

```

OUTPUT:

```

m.akshaya@Akshaya-MacBook-Air ~ % javac myclass.java
m.akshaya@Akshaya-MacBook-Air ~ % java myclass
count:3
value of pi:3.14
value of pi:3.14
value of pi:3.14
M.Akshaya
m.akshaya@Akshaya-MacBook-Air ~ % █

```

ERROR TABLE:

Code Error	Code rectification
------------	--------------------

<ol style="list-style-type: none"> 1. Not Putting the semi-colon after calling a function, 2. Not giving the indentation properly. 	<ol style="list-style-type: none"> 1. Put the semi-colon after calling a function. 2. All the indentation must be correct to run the code correct.
--	--

NEGATIVE CASE:

```

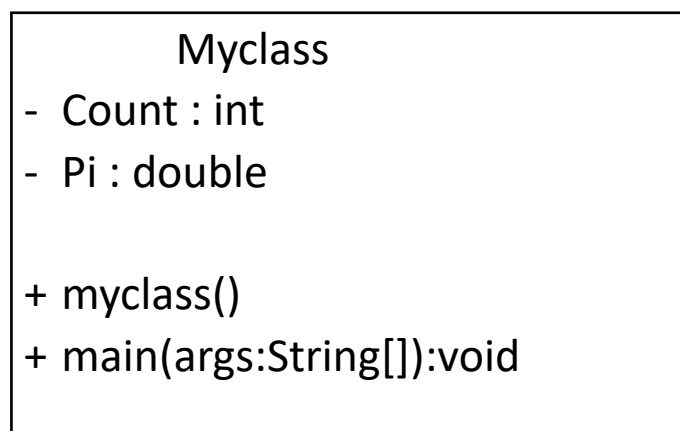
11 errors
[m.akshaya@Akshaya-MacBook-Air ~ % javac myclass.java
myclass.java:2: error: <identifier> expected
    Static int count = 0;
        ^
myclass.java:3: error: <identifier> expected
    Final double pi=3.14;
        ^
2 errors

```

IMPORTANT POINTS:

1. We must declare the initial value of the variable before declaring the final one.
2. Here the main objective is to increase the count according to the number of objects we make, i.e the count increases when the no.of objects are increasing.

CLASS DIAGRAM:



WEEK-5

PROGRAM-1:

AIM: Create a calculator using the operations including addition, subtraction, multiplication, and division using multi-level inheritance and display the desired output.

Hint: collect required variables using super class,
Create each class for a parameter and each class must contain a method.

CODE :

```
class calculator {
    protected double a, b;
    public calculator(double a, double b) {
        this.a = a;
        this.b = b;
    }
}

class Addition extends calculator {
    public Addition(double a, double b) {
        super(a, b);
    }
    public double add() {
        return a + b;
    }
}

class Subtraction extends Addition {
    public Subtraction(double a, double b) {
        super(a, b);
    }
    public double subtract() {
        return a - b;
    }
}
```

```

    }
    class Multiplication extends Subtraction {
        public Multiplication(double a, double b) {
            super(a, b);
        }
        public double multiply() {
            return a * b;
        }
    }
    class Division extends Multiplication {
        public Division(double a, double b) {
            super(a, b);
        }
        public double divide() {
            if (b != 0) {
                return a / b;
            } else {
                System.out.println("Error");
                return Double.NaN;
            }
        }
    }
    class Final extends Division {
        public Final(double a, double b) {
            super(a, b);
        }
        public void displayResults() {
            System.out.println("Addition: " + add());
            System.out.println("Subtraction: " + subtract());
            System.out.println("Multiplication: " + multiply());
            System.out.println("Division: " + divide());
        }
    }
    import java.util.Scanner;
    public class allcalculator {

```

```

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.println("Enter a number: ");
    double a = input.nextDouble();
    System.out.println("Enter b number: ");
    double b = input.nextDouble();
    Final calc = new Final( a, b);
    calc.displayResults();
    System.out.println("Akshaya");
    input.close();
}
}

```

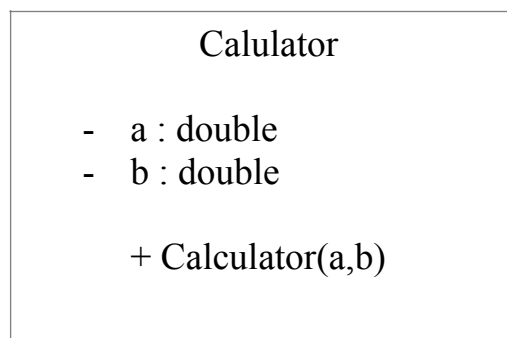
OUTPUT:

```

Enter a number:
65
Enter b number:
24
Addition: 89.0
Subtraction: 41.0
Multiplication: 1560.0
Division: 2.7083333333333335
Akshaya

```

CLASS DIAGRAM:



Addition
+ add(): double

Subtraction
+ subtract(): double

Multiplication
+multiply(): double

Division
+divide(): double



ERROR TABLE:

Code Error	Code rectification
------------	--------------------

<ol style="list-style-type: none"> 1. not providing the return method correctly. 2. Not mentioning super to obtain the super class constructor. 	<ol style="list-style-type: none"> 1. After declaring methods, we must provide the return method correctly. 2. To obtain the super class we need to mention super.
---	--

NEGATIVE CASE:

```
m.akshaya@Akshaya-MacBook-Air ~ % javac allcalculator.java
allcalculator.java:56: error: class, interface, enum, or record expected
import java.util.Scanner;
^
1 error
```

IMPORTANT POINTS:

1. To get the inputs from the user we use import java.util.Scanner; this is a package.
2. Scanner class is used to get the user input.
3. in java.util.Scanner, the java.util is a package while Scanner is a class of the java.util package.
4. to import a whole package, end the sentence with an asterisk sign(*)).

PROGRAM-2

AIM: 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(should be in super class)

1. cars should have an additional property: no.of doors
2. Bikes should have a property indicating whether they have gears or not.

3. The system should also include a function to display details about each vehicle and indicate when a vehicle is starting.
4. Every class should have a constructor

Question:

1. Which oops concept is used in the above program
2. If the company decides to add a new type of vehicle, Truck, how would you modify the program?
 - a. Truck should include an additional property capacity (in tons)
 - b. Create a showTruckdetails() 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 sub classes Finally, display the details.

CODE:

```
public class vehicle {  
    public String brand;  
    public int speed;  
    public vehicle(String brand, int speed) {  
        this.brand = brand;  
        this.speed = speed;  
    }  
    public void start() {  
        System.out.println(brand + " is starting");  
    }  
    public void showDetails() {  
        System.out.println("Brand: " + brand);  
        System.out.println("Speed: " + speed + " km/h");  
    }  
}
```

```

    }
}
class Car extends vehicle {
    private int noOfDoors;
    public Car(String brand, int speed, int noOfDoors) {
        super(brand, speed);
        this.noOfDoors = noOfDoors;
    }
    public void showDetails() {
        super.showDetails();
        System.out.println("Number of Doors: " + noOfDoors);
    }
}
class Bike extends vehicle {
    private boolean hasGears;
    public Bike(String brand, int speed, boolean hasGears) {
        super(brand, speed);
        this.hasGears = hasGears;
    }
    public void showDetails() {
        super.showDetails();
        System.out.println("Has Gears: " + (hasGears ? "Yes" :
"No"));
    }
}
class Truck extends vehicle {
    private int capacity;
    public Truck(String brand, int speed, int capacity) {
        super(brand, speed);
        this.capacity = capacity;
    }
    public void showTruck() {
        super.showDetails();
        System.out.println("Capacity: " + capacity + " tons");
    }
}

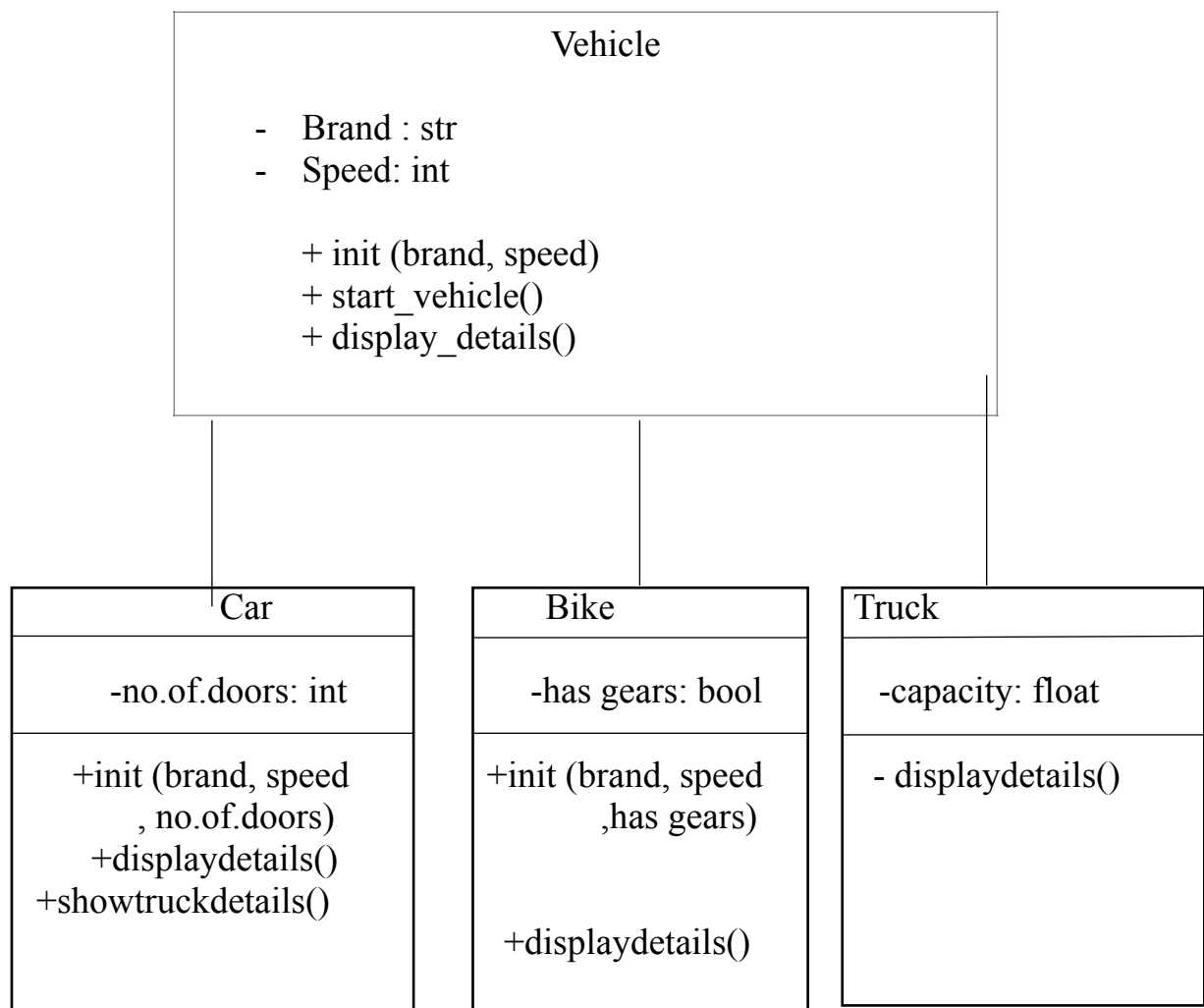
```

}

OUTPUT:

```
Car Details
Toyota is starting
Brand: Toyota
Speed: 150 km/h
Number of Doors: 4
Bike Details
Yamaha is starting
Brand: Yamaha
Speed: 120 km/h
Has Gears: Yes
Truck Details
Volvo is starting
Brand: Volvo
Speed: 90 km/h
Capacity: 10 tons
Akshaya
```

CLASS DIAGRAM:



ERROR TABLE:

Code Error	Code rectification
<ol style="list-style-type: none">1. Declaring two superclasses inside the same file.2. Not declaring the variable using 'this' keyword inside the constructor.	<ol style="list-style-type: none">1. Make two separate files to save the two super classes.2. Declare the variable using this keyword to run the program.

NEGATIVE CASE:

```
● m.akshaya@Akshaya-MacBook-Air ~ % javac vehicle.java
⊗ m.akshaya@Akshaya-MacBook-Air ~ % java vehicle
Error: Main method not found in class vehicle, please define the main method as:
    public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
○ m.akshaya@Akshaya-MacBook-Air ~ % █
```

IMPORTANT POINTS:

1. a constructor helps in initializing an object that doesn't exist.
2. a method performs functions on pre-constructed or already developed objects.
3. a double method can represent more decimal point numbers than float method.
4. the void keyword in java is used to specify that a method does not return any value. it is a return type that indicates the method performs a function and doesn't produce a result.

Answer:

The oops concepts used in the above program are: Inheritance, encapsulation, polymorphism, abstraction. To add a new vehicle type truck we need to create a truck class that will:

- Include an additional property capacity (in tons).
- Implement a showtruckdetials() method to display the truck's capacity.
- Implement a constructor for the truck class to initialize all its properties.

WEEK-6:

PROGRAM-1

AIM:

CODE:

```
class vehicle {
    public String car_model;
    public String color;
    public String fuel_type;
    Public vehicle (String car_model, String color, String
fuel_type) {
        this.car_model = car_model;
        this.color = color;
        this.fuel_type = fuel_type;
    }
    void displayDetails() {
        System.out.println("car_model: " + car_model);
        System.out.println("car_color: " + color);
        System.out.println("car_fuel_type: " + fuel_type);
        System.out.println("I have an Porche");
    }
}

class Car extends vehicle {
    public Car (String car_model, String color, String fuel_type) {
```

```

        super(car_model,color,fuel_type);
    }
    @Override
    void displayDetails() {
        super.displayDetails();
        System.out.println("I have a ferrari ");
    }
}
public class Truck {
    public static void main(String[] args) {
        vehicle v = new vehicle("Porche","royal blue","diesel");
        v.displayDetails();
        Car c = new Car ("ferrari","red","petrol");
        c.displayDetails();
    }
}

```

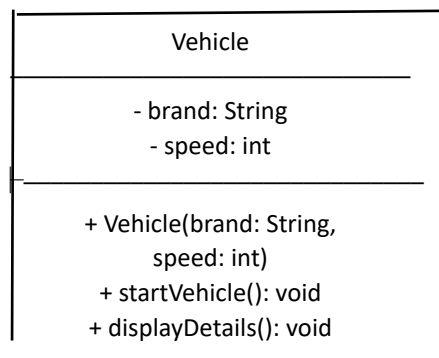
OUTPUT:

```

m.akshaya@Akshaya-MacBook-Air MyProject % java Truck
car_model: Porche
car_color: royal blue
car_fuel_type: diesel
I have an Porche
car_model: ferrari
car_color: red
car_fuel_type: petrol
I have an Porche
I have a ferrari

```

CLASS DIAGRAM:



ERROR TABLE:

Error	Error Rectification
1.Incorrect class name for main method (Truck)	1.Rename Truck to Main or place main inside Car or Vehicle.
2.Inconsistent car model output in displayInfo()	2.Ensure Car correctly passes "Benz" to super(car_model, color, fuel_type);.

NEGATIVE CASE:

```
caused by: java.lang.ClassNotFoundException: Vehicle
m.akshaya@Akshaya-MacBook-Air MyProject % java vehicle
Error: Main method not found in class vehicle, please define the main method as:
    public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
```

IMPORTANT POINTS:

Inheritance: The Car class extends the Vehicle class, demonstrating **inheritance** in Java.

Constructor Chaining: The Car class calls the parent constructor using super(car_model, color, fuel_type); to initialize inherited attributes.

Method Overriding: The Car class overrides the displayInfo() method from Vehicle and calls super.displayInfo() to reuse the parent method before adding its own output.

Incorrect main Class Name:The main method is inside Truck, which is unrelated to Vehicle and Car. The class should be renamed for clarity.

PROGRAM-3:

```
public 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;  
    }  
    public static void main (String[] args) {  
        calculator calc = new calculator();  
        System.out.println("sum: " + calc.add(20,50));  
        System.out.println("sum of 3.5 and 6.2:" +  
calc.add(3.5,6.2));  
        System.out.println("sum of 20,30 and 20: " +  
calc.add(20,30,20));  
    }  
}
```

OUTPUT:

```
m.akshaya@Akshaya-MacBook-Air MyProjects4 % javac calculator.java  
m.akshaya@Akshaya-MacBook-Air MyProjects4 % java calculator  
sum: 70  
sum of 3.5 and 6.2:9.7  
sum of 20,30 and 20: 70
```

CLASS DIAGRAM:

Calculator
+add(int, int): int +add(double, double): double +add(int, int, int): int +main(String[]): void

ERROR TABLE:

Code Errors	CODE RECTIFICATIONS
1.Method parameters missing spaces (e.g., "int a,int b" should be "int a, int b") 2. Inconsistent indentation in method bodies (some lines not properly aligned)	1.Add proper spacing between Parameters: -old:add(int a,intb) -New:add(int a,int b) 2.Fix indentation: consist 4 space of indentation

NEGATIVE CASE:

```

m.akshaya@Akshaya-MacBook-Air ~ % cd MyProjects4
m.akshaya@Akshaya-MacBook-Air MyProjects4 % javac calculator.java
calculator.java:13: error: ')' or ',' expected
    System.out.println("sum: " + calc add(20,50));
                                   ^
calculator.java:13: error: ';' expected
    System.out.println("sum: " + calc add(20,50));
                                   ^

```

IMPORTANT POINTS:

1.**Method Overloading:** The add method is overloaded with different parameter types and counts, demonstrating compile-time polymorphism.

2.**Automatic Method Selection:** Java selects the appropriate add method based on the argument types during compilation.

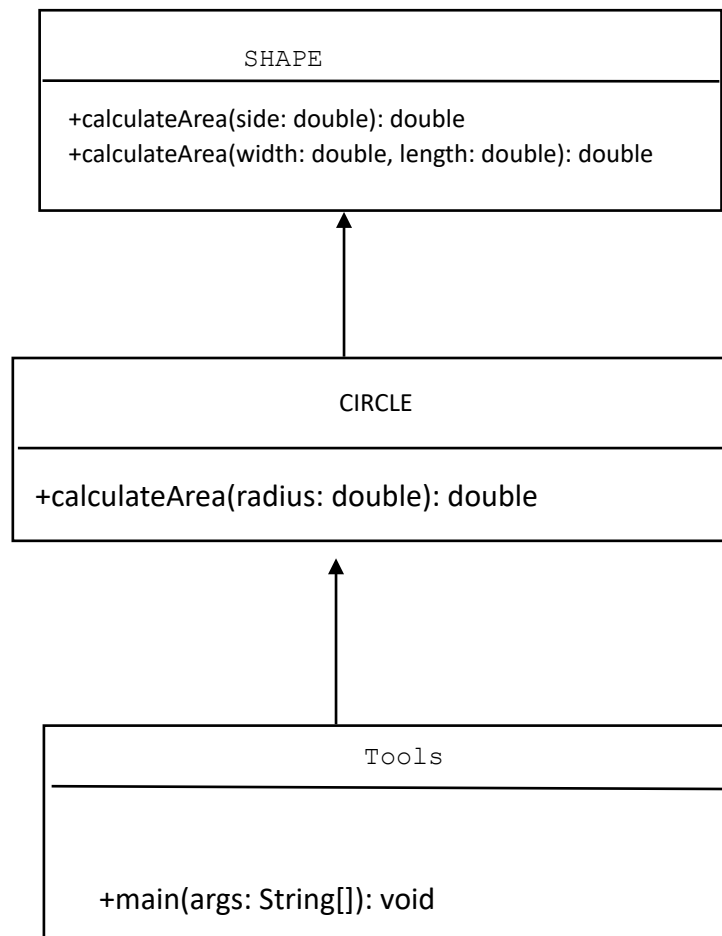
Program-2

```
class shape {
    double calculateArea(double side) {
        return side * side ;
    }
    double calculateArea(double width , double length) {
        return width * length ;
    }
}
class Circle extends shape {
    double calculateArea(double radius) {
        return 3.14 * radius * radius ;
    }
}
public class Tools {
    public static void main(String[] args) {
        shape s = new shape();
        Circle c = new Circle();
        System.out.println("Area of square (side 6): " +
s.calculateArea(6));
        System.out.println("Area of rectangle (6x5): " +
s.calculateArea(6,5));
        System.out.println("Area of circle (radius 4): " +
s.calculateArea(4));
    }
}
```

OUTPUT:

```
m.akshaya@Akshaya-MacBook-Air MyProjects4 % javac Tools.java
m.akshaya@Akshaya-MacBook-Air MyProjects4 % java Tools
Area of square (side 6): 36.0
Area of rectangle (6x5): 30.0
Area of circle (radius 4): 16.0
```

Class Diagram:



ERROR TABLE:

Error	Error Rectification
1. Method calls in main are missing an object reference (e.g., <code>calculateArea(4)</code> instead of <code>s.calculateArea(4)</code>).	Use <code>s.calculateArea(4)</code> and <code>c.calculateArea(2)</code> to call the method correctly.
2.Circle class method does not override the parent class method properly.	2.Ensure <code>@Override</code> is used, and the method signature should match correctly.

Negative case:

```
m.akshaya@Akshaya-MacBook-Air MyProjects4 % javac Tools.java
Tools.java:15: error: <identifier> expected
    public static void main(String[] args) {
                   ^
1 error
```


Important Points:

- 1.Inheritance:** Circle class extends Shape, inheriting its methods.
- 2.Method Overloading:** Shape has multiple calculateArea methods with different parameters.
- 3.Method Overriding:** Circle overrides calculateArea from Shape to implement its own formula.
- 4.Polymorphism:** The overridden method in Circle demonstrates runtime polymorphism.
- 5.Proper Object Reference:** Methods should be called using an object (s.calculateArea(4), c.calculateArea(2)).

PROGRAM-4:

```
import java.util.Scanner;
```

```
public class AdmissionSystem {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.println("Enter student name:");  
        String name = scanner.nextLine();  
  
        System.out.println("Enter previous qualification  
percentage:");  
        double percentage = scanner.nextDouble();  
  
        System.out.println("Enter program type (UG/PG):");  
        String program = scanner.next().toUpperCase();  
  
        if (program.equals("UG")) {  
            if (percentage >= 60) {  
                System.out.println("Congratulations, " + name + "! You  
are eligible for UG admission.");  
            }  
        }  
    }  
}
```

```

        } else {
            System.out.println("Sorry, " + name + ". You do not
meet the UG admission criteria.");
        }
    } else if (program.equals("PG")) {
        if (percentage >= 70) {
            System.out.println("Congratulations, " + name + "! You
are eligible for PG admission.");
        } else {
            System.out.println("Sorry, " + name + ". You do not
meet the PG admission criteria.");
        }
    } else {
        System.out.println("Invalid program type entered. Please
enter UG or PG.");
    }
    scanner.close();
}
}

```

OUTPUT:

```

[m.akshaya@Akshaya-MacBook-Air MyProject % javac AdmissionSystem.java
[m.akshaya@Akshaya-MacBook-Air MyProject % java AdmissionSystem
Enter student name:
Aryaan
Enter previous qualification percentage:
75
Enter program type (UG/PG):
UG
Congratulations, Aryaan! You are eligible for UG admission.

```

CLASS DIAGRAM:

AdmissionSystem
- scanner: Scanner - name: String - percentage: double - program: String
+ main(args: String[]): void + takeInput(): void + checkEligibility(): void + closeScanner(): void

Error Table:

Error	Error Rectification
1. Scanner <code>nextLine()</code> issue after <code>nextDouble()</code> : After <code>scanner.nextDouble()</code> , the newline character remains in the buffer, causing <code>nextLine()</code> to be skipped.	1. Add <code>scanner.nextLine()</code> ; after <code>nextDouble()</code> ; to consume the leftover newline.
2. Program type input case sensitivity issue: If the user enters ug or pg in lowercase, it may cause incorrect comparisons.	2. Use <code>program.toUpperCase()</code> to ensure case-insensitive comparison.

NEGATIVE CASE:

```
C:\Users\svehy\OneDrive\Documents\MATLAB\javascriptusinghtml>JAVAC ADMISSIONSYAYTEM
error: Class names, 'ADMISSIONSYAYTEM', are only accepted if annotation processing is explicitly requested
1 error
C:\Users\svehy\OneDrive\Documents\MATLAB\javascriptusinghtml>
```

Important Points:

User Input Handling: Uses Scanner to take user input for name, percentage, and program type.

Decision Making with Conditions: Uses if-else statements to check eligibility criteria.

String Handling: Converts program input to uppercase (`toUpperCase()`) to handle case variations.

Closing Scanner: Properly closes scanner using `scanner.close()`; to prevent resource leaks.