23CSE111

OBJECT ORIENTED PROGRAMMING

LAB REPORT

Department of Computer Science Engineering **Amrita School of Computing**

Amrita Vishwa Vidyapeetham, Amaravati Campus

Name: PVs krishnakanth

Roll No: 24304

Verified By :

Index

INDEX

Week 1

WEEK-1

1) Process of Installing JDK (Java Development Kit)

Installing JDK (Java Development Kit):

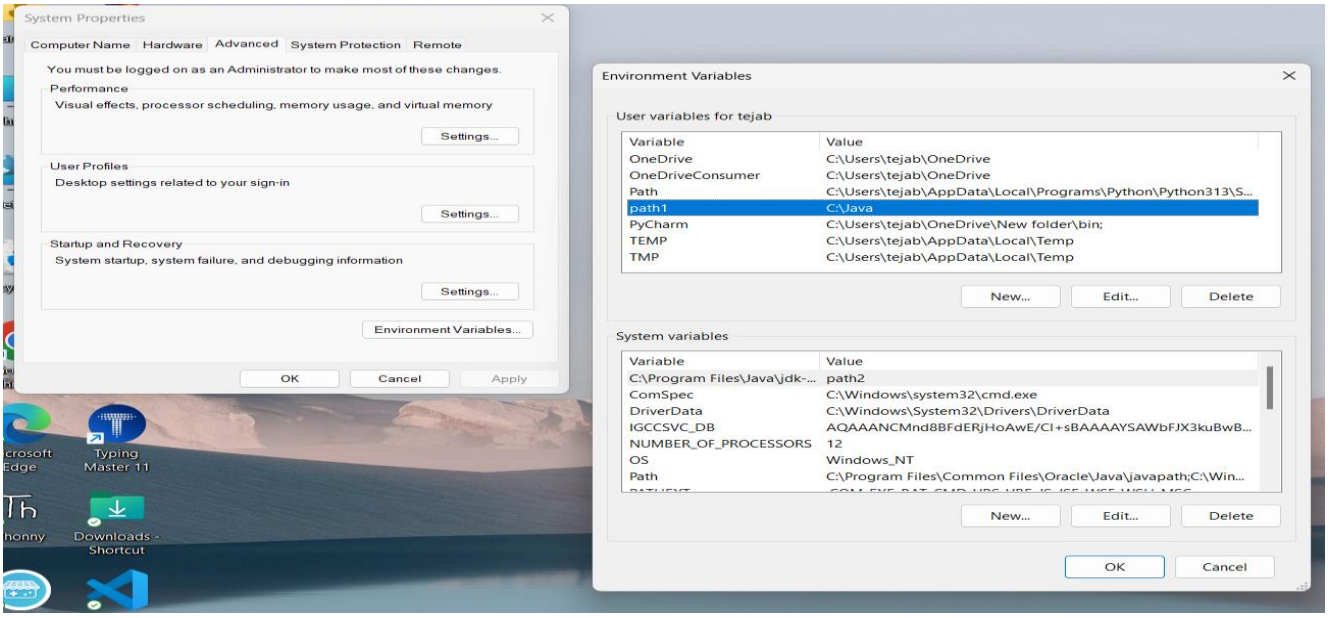
1. **Download JDK:**
   - Go to the Oracle JDK download page in google and click on JDK-21 version which is Long term support (LTS) version.
   - Click the download link as your operating system (Windows, macOS, or Linux).
2. **Install JDK:**
   - Once downloaded, run the installer.
   - Follow the given instructions and keep clicking "Next" until it is 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 on path and new under **System Variables**:

   Variable value: The folder address where JDK is installed (like C:\Program Files\Java\jdk-21\bin)

   - Find Path under **System Variables**, click **New**, and add the path of the jdk-21(C:\Program Files\Java\jdk-21\bin)



Checking 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.

2) Simple Java Program for printing Name, Class, Roll No, of a Student

Aim:

Write your code in Notepad and execute it in cmd prompt

CODE: -

```
class Main
{
	public static void main(String[] args)
		{
			System.out.println("Name: pvs krishnakanth");
			System.out.println("Class :CSE-b");
			System.out.println("Roll No:24304");
		}
}
```

Output: -

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\java labmanual>javac main.java

C:\java labmanual>java main.java
Name: pvs krishnakanth
Class :CSE-b
Roll No:24304

C:\java labmanual>
```

Errors

1	Syntax error	Semicolon added	
2	Runtime error	Copied correct path	
3	Name error	rectified	

Negative Case:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\java labmanual>javac Helloworld.java
Helloworld.java:1: error: class HelloWorld is public, should be declared in a file named HelloWorld.java
public class HelloWorld {
       ^
1 error
```

Week-2

1) Write a Simple Java Program for finding simple interest by taking input

```
Simport java.util.*;

class Simple {
   public static void main(String args[]) {
      float SI;
      Scanner sc = new Scanner(System.in);

      System.out.println("Enter principal value:");
      int P = sc.nextInt();

      System.out.println("Enter time:");
      int T = sc.nextInt();

      System.out.println("Enter rate of interest:");
      float R = sc.nextFloat();

      SI = (P * T * R) / 100;
      System.out.println("Simple interest is: " + SI);
   }
}
```

Output:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\java labmanual>javac Simple.java

C:\java labmanual>java Simple.java
Enter principal value:
455
Enter time:
5
Enter rate of interest:
2
Simple interest is: 45.5
```

Negative Case:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\java labmanual>javac Simple.java
Simple.java:1: error: class, interface, enum, or record expected
Simport java.util.*;
^
Simple.java:17: error: ';' expected
        SI = (P * T * R) / 100
                           ^
2 errors

C:\java labmanual>
```

Error Table:

S.No	Error type	Reason for error	rectification
1	Runtime error	Incorrect path	Copied correct path

2	**Syntax error**	**{ missing**	**{ added**
3	**Logical error**	**Wrong formula**	**Formula rectified**

2.Aim: Write a simple java program to calculate factorial of a number and read the input from user

code:

```
class fac{
  public static void main(String[] args){
        int fac=1;
        int n=10;
        for(int i=1;i<=n;i+=1){
              fac*=I;
}
         System.out.println("fac of first 10 natural numbers="+fac);



}

}
```

Output:

Negative case:

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\java labmanual>javac fac.java
fac.java:6: error: ';' expected
                    fac*=i
                          ^
1 error

C:\java labmanual>
```

```
C:\Users\tejab>cd C:\java files\Lab files java\Lab 2

C:\java files\Lab files java\Lab 2>javac Fact.java
Fact.java:14: error: ';' expected
    static int factorial(int n)
                              ^
Fact.java:18: error: illegal start of type
        return n * factorial(n - 1);
        ^
Fact.java:18: error: <identifier> expected
        return n * factorial(n - 1);
                 ^
Fact.java:18: error: invalid method declaration; return type required
        return n * factorial(n - 1);
                   ^
Fact.java:18: error: <identifier> expected
        return n * factorial(n - 1);
                             ^
Fact.java:21: error: class, interface, enum, or record expected
}
^
6 errors
```

Error table

S.No	Error type	Reason for error	Rectification
1	Undeclared variable error	Missing variable	Variable declared
2	Missing import statement	Not importing packages	Packages imported
3	Logical error	Wrong formula	Formula rectified

3.

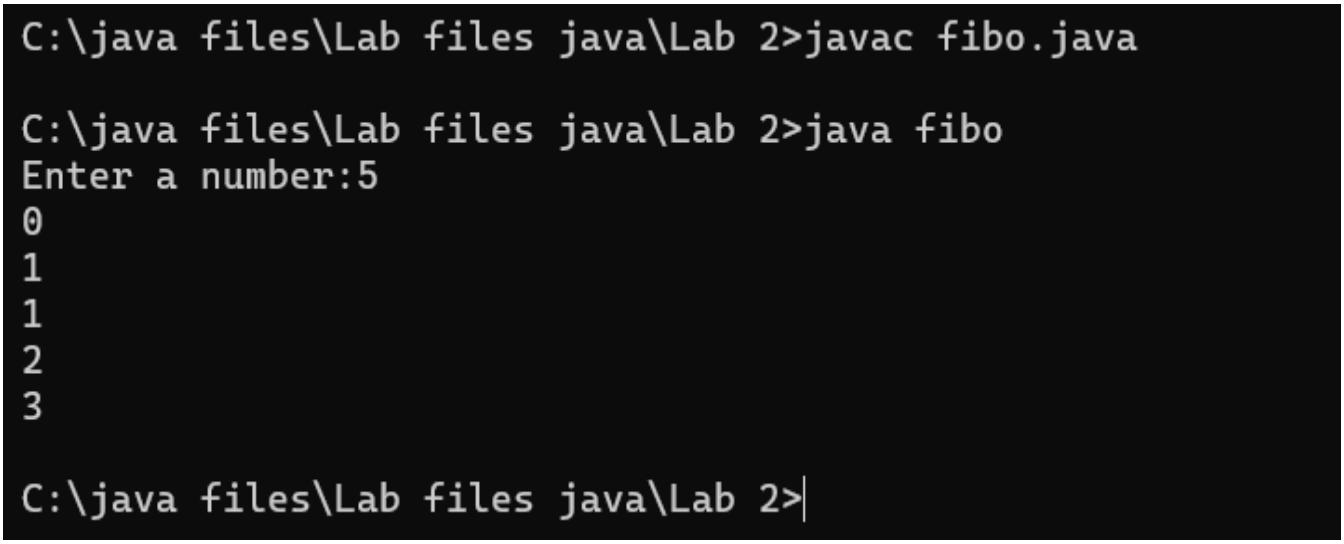
Aim: Write a program to to calculate the fibonacii sequence and take the input from user

Code:

```
import java.util.*;
class fibo
{
  public static void main(String args[])
 {
    Scanner sc = new Scanner(System.in);
    int num;
    int f3;
    int f1 = 0;
    int f2 = 1;
    int i = 2;

    System.out.print("Enter a number:");
    num = sc.nextInt();
      System.out.println(f1);
    System.out.println(f2);

    while(i<num)
    {
```

```
        f3 = f1+f2;
        f1 = f2;
        f2 = f3;
        System.out.println(f3);
        i = i+1;
    }
  }
}
```

Output:



Error Table:

S.No	Error type	Reason for error	Rectification
1	Logical error	Incorrect formula	Formula rectified
2	Run-time error	Incorrect path	Added correct path
3			

Negative case:

```
C:\java files\Lab files java\Lab 2>javac fibo.java
fibo.java:13: error: cannot find symbol
      System.outprint("Enter a number:");
            ^
  symbol:   method outprint(String)
  location: class System
1 error
```

Important points

Here the assignment operartion takes makes values are keep on updated for f1 and f2

4.A)

Aim: Write a java program to convert temperature from Fahrenheit to celsius

Code:

```
import java.util.*;
class temp
{
     public static void main(String args[])
     {
         Scanner sc = new Scanner(System.in);
         float c;
         System.out.println("Enter fahrenheit temperature:");
         float f = sc.nextFloat();
         c = (f-32)*5/9;
         System.out.println("Celsius Temperature is :"+c);
      }
}
```

Output:

```
C:\java files\Lab files java\Lab 2>javac temp.java

C:\java files\Lab files java\Lab 2>java temp
Enter fahrenheit temperature:
98.6
Celsius Temperature is :37.0

C:\java files\Lab files java\Lab 2>
```

Negative Case:

```
C:\java files\Lab files java\Lab 2>javac temp.java
temp.java:10: error: cannot find symbol
         c = (F-32)*5/9;
              ^
  symbol:   variable F
  location: class temp
1 error
```

Error Table:

S.No	Error type	Reason for error	Rectification
1	Syntax error	Missing ”	“ is added
2	Missing import error	Util package missing	Util package added
3	Logical error	Incorrect formula	Formula rectified

Important Points

Conversion of Fahrenheit to Celsius is c = (f-32)*5/9

4.b

Aim: Write a java program to convert temperature from Celsius to Fahrenheit

Code:

```
    import java.util.*;
class heat
{
     public static void main(String args[])
     {
         Scanner sc = new Scanner(System.in);
         float f;
         System.out.println("Enter celsius  temperature:");
         float c = sc.nextFloat();
         f = (c*9/5)+32;
         System.out.println("Fahrenheit  temperatue is :"+f);
      }
}
```

Output:

```
C:\java files\Lab files java\Lab 2>javac heat.java

C:\java files\Lab files java\Lab 2>java heat
Enter celsius  temperature:
34
Fahrenheit temperatue is :93.2
```

Negative Case:

```
C:\java files\Lab files java\Lab 2>javac heat.java
heat.java:1: error: <identifier> expected
    import java.util.;
                     ^
1 error

C:\java files\Lab files java\Lab 2>
```

Important points

Conversion of celsius to Fahrenheit is (c*9/5)+32

Error Table:

S.No	Error type	Reason for error	Rectification
1	Runtime error	Incorrect path selection	Correct path added
2	Logical error	Incorrect formula	Correct formula rectified
3	Import package error	Incorrect importing of Packages	Imported util.*; Package

5.

Aim: Write a simple java program to find the area of rectangle:

Code:

```
import java.util.*;
class area
{
   public static void main(String args[])
   {
         int area;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter length:");
        int l = sc.nextInt();
        System.out.println("Enter breadth:");
        int b = sc.nextInt();
        area = l*b;
        System.out.println("Area of rectangle:"+area);
   }
}
```

Output:

```
C:\Users\tejab>cd C:\java files\Lab files java\Lab 2

C:\java files\Lab files java\Lab 2>javac area.java

C:\java files\Lab files java\Lab 2>java area
Enter length:
5
Enter breadth:
2
Area of rectangle:10

C:\java files\Lab files java\Lab 2>
```

Error Table:

S.No	Error type	Reason for error	Rectification
1	Syntax error	Semi colon missing	Semi colon added
2	Missing import error	Import package missing	Import package added
3	Runtime error	Incorrect path selection	Rectified correct path

Negative case:

```
C:\java files\Lab files java\Lab 2>javac area.java

C:\java files\Lab files java\Lab 2>java area
Enter length:
1
Enter breadth:
w
Exception in thread "main" java.util.InputMismatchException
        at java.base/java.util.Scanner.throwFor(Scanner.java:964)
        at java.base/java.util.Scanner.next(Scanner.java:1619)
        at java.base/java.util.Scanner.nextInt(Scanner.java:2284)
        at java.base/java.util.Scanner.nextInt(Scanner.java:2238)
        at area.main(area.java:11)
```

6.Aim:

Write a program to find the area of triangle by using heron's formula take the input from the user

Code:

```
import java.util.*;
import java.lang.Math;
class heron
{
     public static void main(String args[])
     {

          double s, c,a,b,p;
          Scanner sc = new Scanner(System.in);
          System.out.println("Enter the values of a , b and c:");
           a = sc.nextDouble();
           b = sc.nextDouble();
           c = sc.nextDouble();
          s = (a+b+c)/2;
          p = Math.sqrt(s*(s-a)*(s-b)*(s-c));
          System.out.println(" Area of triangle by heron's formula is :"+p);
     }
}
```

OUTPUT:

```
C:\java files\Lab files java\Lab 2>javac heron.java

C:\java files\Lab files java\Lab 2>java heron
Enter the values of a , b and c:
12
10
12
 Area of triangle by heron's formula is :54.543560573178574

C:\java files\Lab files java\Lab 2>
```

Negative Case:

```
C:\java files\Lab files java\Lab 2>javac heron.java
heron.java:2: error: <identifier> expected
import java.lang.
                ^
1 error

C:\java files\Lab files java\Lab 2>
```

Error Table

S.No	Error type	Reason for error	Rectification	
1	Logical error	Incorrect formula	Formula rectified	
2	Name error	Undeclared variable	Variable declared	
3	Import package error	Incorrect package	Package Recttified	

Important points

Import java.lang.Math is used to access the built in Math Class which provides a collection

of static methods

For performing various mathematical operations

Week-3

WEEK 3

1. **Aim:**

To create java program with following instructions

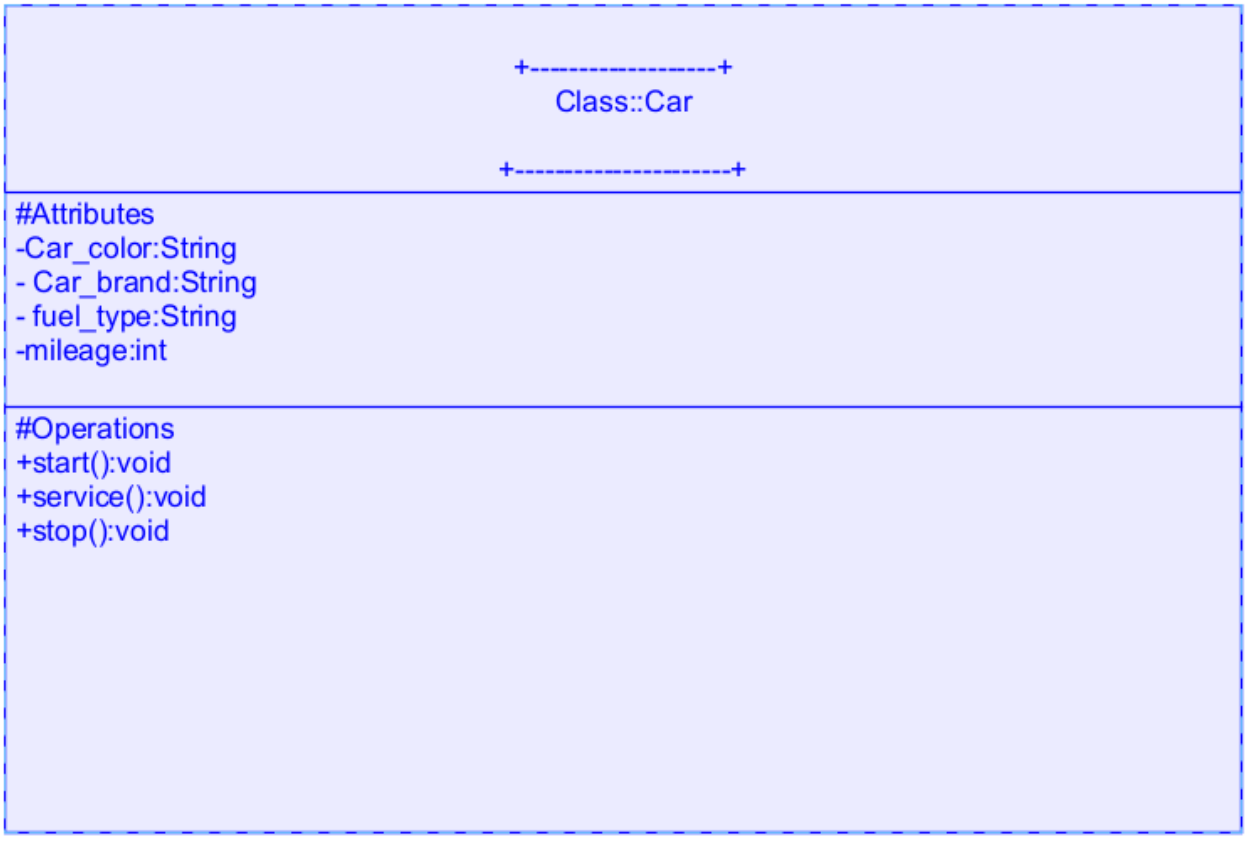
1.Create a class with name car

2. Create four attributes named car_color ,Car_brand,fuel_type,mileage

3. Create three methods named start(), stop(). Service()

4. Create three objects named car1,car2 and car3

Class Diagram:



Code:

```
import java.util.*;
class car
{

   public String Car_color;
   public String Car_brand;
   public String fuel_type;
   public int mileage;
  public void start()
  {

     System.out.println("Car Started:");
     System.out.println("Car color is :"+Car_color);
     System.out.println("Car Brand is:"+Car_brand);
     System.out.println("Car fuel type is:"+fuel_type);
     System.out.println("Car mileage is:"+mileage);
  }
```

```
 public void service()

{
   System.out.println("Car Started:");
   System.out.println("Car color is :"+Car_color);
   System.out.println("Car Brand is:"+Car_brand);
   System.out.println("Car fuel type is:"+fuel_type);
   System.out.println("Car mileage is:"+mileage);
}
 public void stop()

{
   System.out.println("Car Started:");
   System.out.println("Car color is :"+Car_color);
   System.out.println("Car Brand is:"+Car_brand);
   System.out.println("Car fuel type is:"+fuel_type);
   System.out.println("Car mileage is:"+mileage);
}
public static void main(String args[])
{   System.out.println("\nBHANU TEJA\n\n");
   car car1 = new car();
   car1.Car_color = "Blue";
   car1.Car_brand = "Audi";
   car1.fuel_type = "Deisel";
   car1.mileage = 100;
```

```
      car1.start();
       car car2 = new car();
       car2.Car_color = "Red";
       car2.Car_brand = "Tesla";
       car2.fuel_type = "EV";
       car2.mileage = 200;
       car2.stop();
       car car3 = new car();
       car3.Car_color = "Yellow";
       car3.Car_brand = "BMW";
       car3.fuel_type = "Petrol";
       car3.mileage = 300;
      car3.service();


    }
}
```

Important points

Methods: The set of instructions that can be called for execution using a method name.

Output:

```
C:\java files\Lab files java\Lab 2>javac car.java

C:\java files\Lab files java\Lab 2>java car
BHANU TEJA



Car Started:
Car color is :Blue
Car Brand is:Audi
Car fuel type is:Deisel
Car mileage is:100
Car Started:
Car color is :Red
Car Brand is:Tesla
Car fuel type is:EV
Car mileage is:200
Car Started:
Car color is :Yellow
Car Brand is:BMW
Car fuel type is:Petrol
Car mileage is:300

C:\java files\Lab files java\Lab 2>
```

Error Table

S No	Error Type	Cause of error	Rectification
1	Syntax Error	Missing ‘{‘	‘{‘ added
2	Compile time Error	Mispelled Variable call	Rectified with Correct variable name
3			

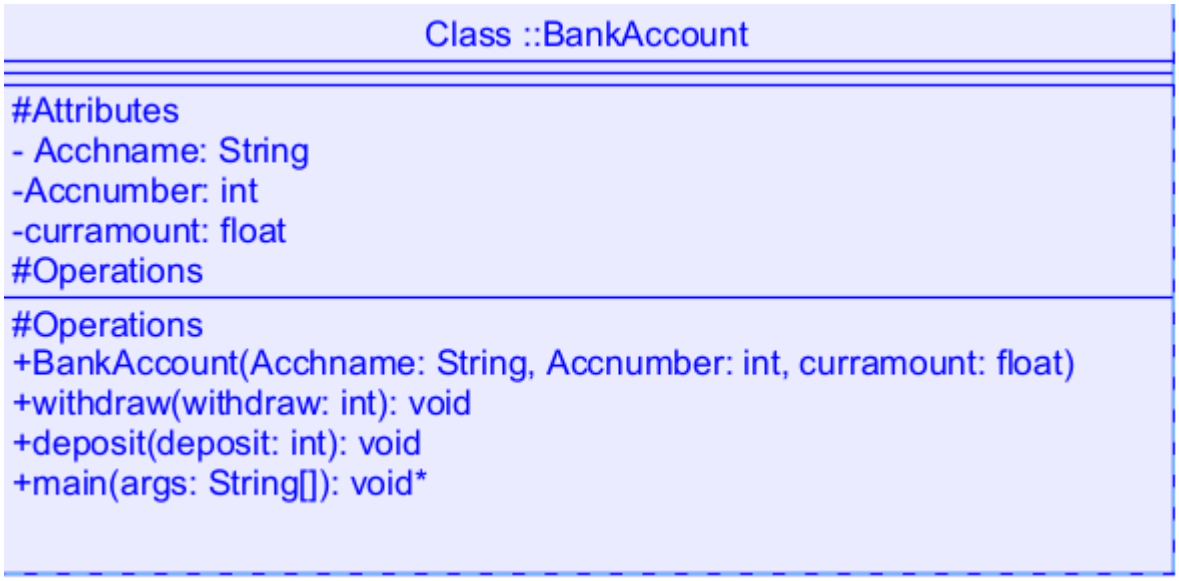
Negative Case

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

C:\Users\tejab>cd C:\java files\Lab files java\Lab 2

C:\java files\Lab files java\Lab 2>javac car.java
car.java:43: error: non-static method start() cannot be referenced from a static context
      car.start();
          ^
1 error

C:\java files\Lab files java\Lab 2>
```

2.

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

Class Diagram



Code:

```
class BankAccount
{

    public String Acchname;
    public int Accnumber;
    public float curramount;
   BankAccount(String Acchname, int Accnum,float curramount)
   {
     this.Acchname = Acchname;
     this.Accnumber = Accnumber;
     this.curramount = curramount;
    System.out.println("Enter Account holder name:"+Acchname);
      System.out.println("Enter Account number:"+Accnum);
      System.out.println("Enter current amount:"+curramount);
   }
   public void withdraw(int withdraw)
   {
     if(withdraw>curramount)
     {
          System.out.println("Insufficient Funds");
     }
     else
     {
         curramount = curramount-withdraw;
```

```
        System.out.println("withdraw amount is:"+withdraw);
        System.out.println("Current amount is:"+curramount);
    }
  }
  public void deposit(int deposit)
    {
        System.out.println("Deposited amount is :");
        curramount = curramount+deposit;
        System.out.println("Deposited amount is:"+deposit);
        System.out.println("Total current amount is:"+curramount);
    }

    public static void main(String args[])
    {

        BankAccount b = new BankAccount("Hari",24210,100000);
        b.deposit(10000);
        b.withdraw(500);

}}
```

Output:

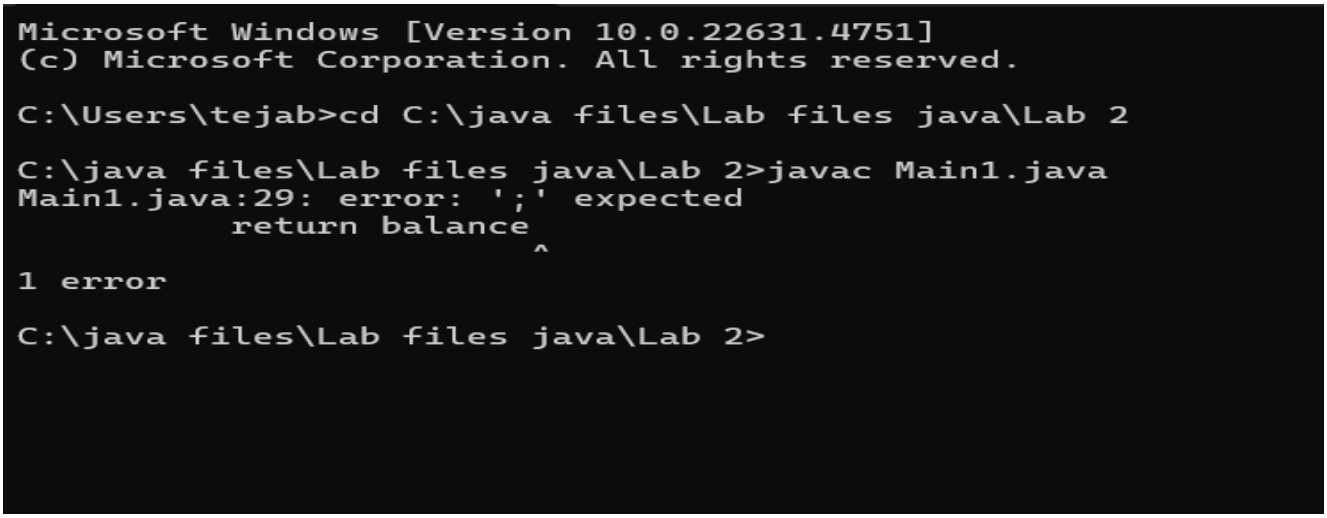
```
C:\Users\tejab>cd C:\java files\Lab files java\Lab 2

C:\java files\Lab files java\Lab 2>javac BankAccount.java

C:\java files\Lab files java\Lab 2>java BankAccount
Enter Account holder name:Hari
Enter Account number:24210
Enter current amount:100000.0
Deposited amount is :
Deposited amount is:10000
Total current amount is:110000.0
withdraw amount is:500
Current amount is:109500.0

C:\java files\Lab files java\Lab 2>
```

Error Table:

s.no	Error name	Cause of error	Rectification
1	Name Error	Undefined name	Correct variable Name replaced
2	Syntax Error	Missing Parenthesis	Parenthesis Added

3	**Logical Error**	**Incorrect Condition**	**Condition Rectified**

Negative Case



Important points:

Constructor: The Constructor creates and initializes objects of a class. They are called when an object is created to a class.

This Keyword: The This keyword refers to the current instance of a class.It is used to Access class variables and methods.

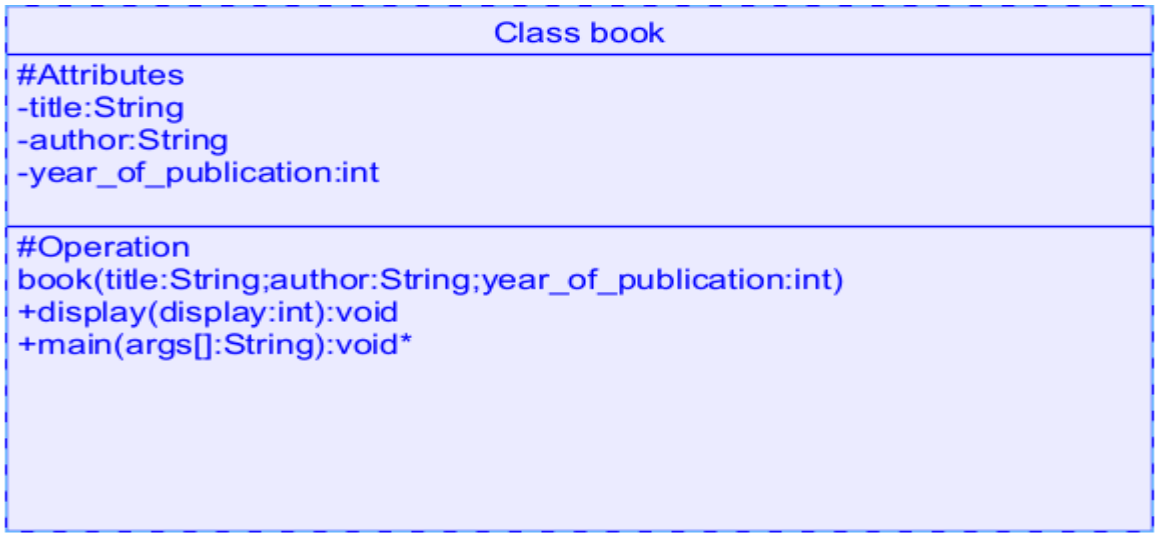
Class Diagram

WEEK 4

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 .Display The details of two books

Class Diagram



Code

```
class book
{
    public String title;
    public String author;
    public int year_of_publication;

    book(String title, String author, int year_of_publication)
    {
        this.title = title;
        this.author = author;
        this.year_of_publication = year_of_publication;

    }
    public void display()
    {
```

```
        System.out.println("Title of book is:"+title);
        System.out.println("Author of book is:"+author);
        System.out.println("Year of publication is:"+year_of_publication);
    }
  public static void main(String args[])
   {
      book b1 = new book("Python","Bhanu",2023);
      b1.display();
      book b2 = new book("Java","Sushanth",2024);
      b2.display();
    }}
```

Output:

```
C:\java files\Lab files java\Lab 4>javac book.java

C:\java files\Lab files java\Lab 4>java book
Title of book is:Python
Author of book is:Bhanu
Year of publication is:2023
Title of book is:Java
Author of book is:Sushanth
Year of publication is:2024
```

Error Table:

S.No	Error Type	Reason	Rectification	
1	Syntax error	Missing {	Added {	
2	Static method called non statically	Fault in calling the method	Rectified method Calling	
3	Run-time error	Incorrect Selection of	Correct path	

		path	**Selected**

Negative Case

```
C:\java files\Lab files java\Lab 4>javac book.java
book.java:23: error: cannot find symbol
          b.display();
          ^
  symbol:   variable b
  location: class book
1 error

C:\java files\Lab files java\Lab 4>
```

2.Create a java program with class named “myclass” with a static variable

“count” of int type, initialized to zero and a constant variable “pi” of type

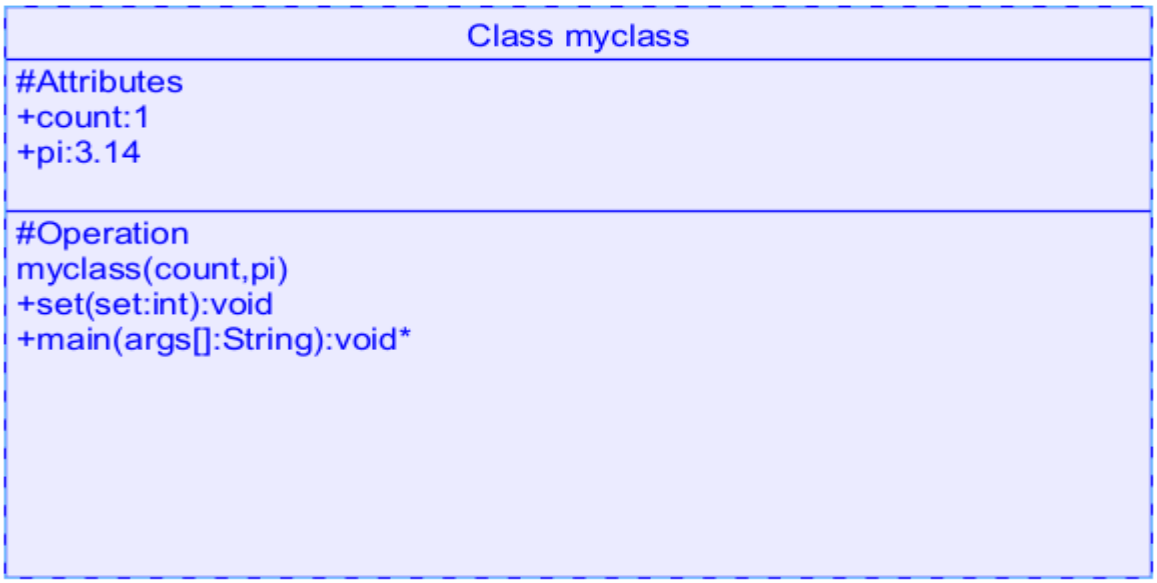
Double initialized to 3.14 as attributes of the class. Now define a constructor

For “myclass” that increments the count variable each time an object of

“myclass” is created

Finally Print the final values of count and pi variables. Create three objects

Class Diagram:

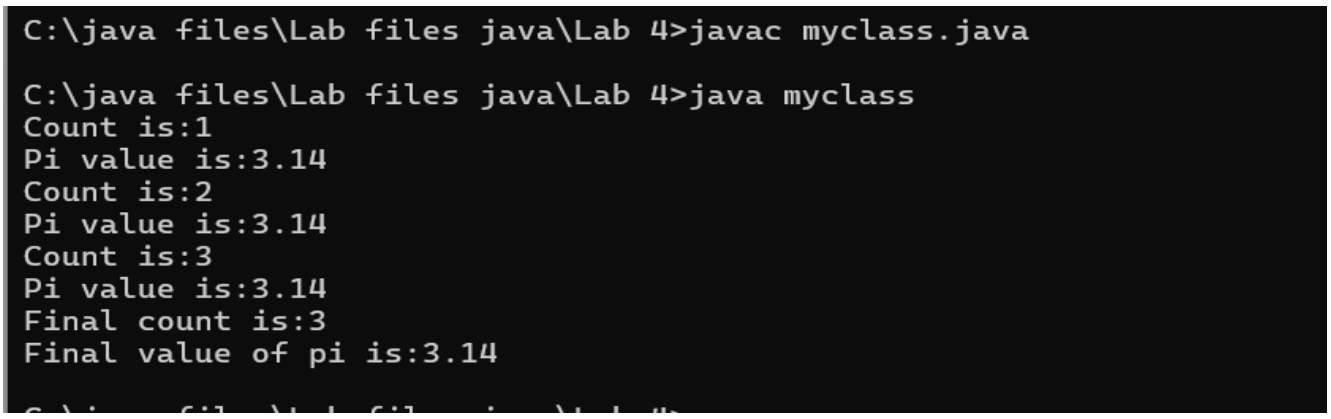


Code:

```
class myclass
{
    static int count=0;
    static double pi=3.14;
    myclass()
    {
       count = count+1;
     }
     public void set()
     {
         System.out.println("Count is:"+count);
         System.out.println("Pi value is:"+pi);
      }
      public static void main(String args[])
      {
```

```
        myclass m = new myclass();
        m.set();
        myclass m1 = new myclass();
        m1.set();
        myclass m2 = new myclass();
        m2.set();
    System.out.println("Final count is:"+count);
    System.out.println("Final value of pi is:"+pi);
   }
}
```

Output:



Negative Case:

```
C:\java files\Lab files java\Lab 4>javac myclass.java
myclass.java:21: error: ';' expected
            m2.set()
                    ^
1 error

C:\java files\Lab files java\Lab 4>
```

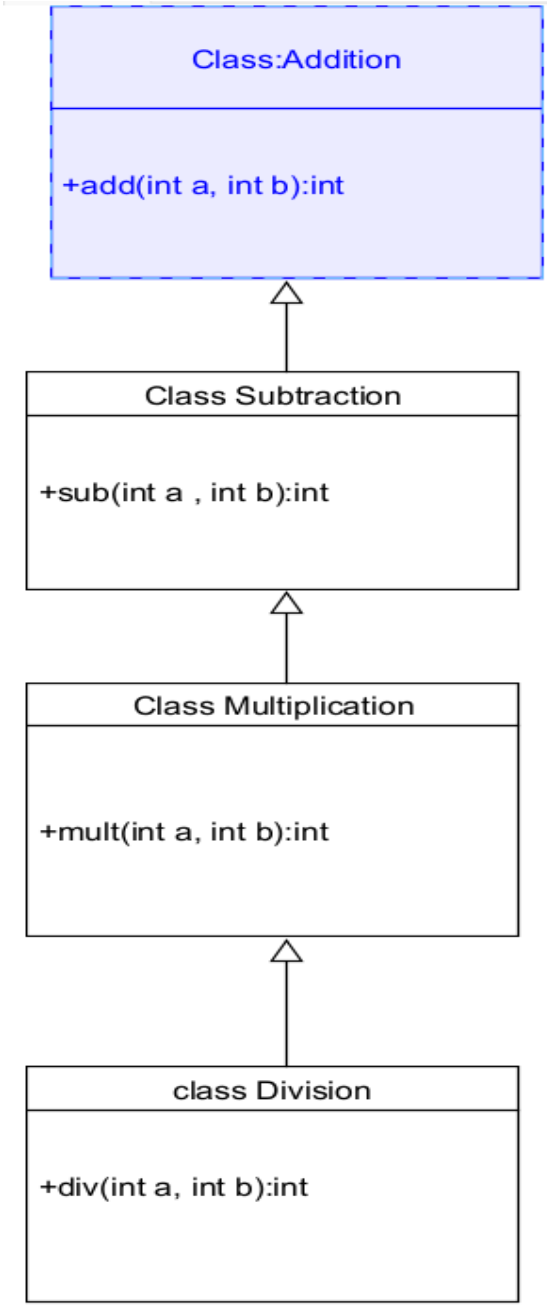
Error Table:

S.No	Error Type	Cause	Rectification
1	Name error	Incorrect variable called	Rectified with correct variable
2	Syntax error	Missing semi-colon	Semi-colon added
3	Run time error	Incorrect path	Selected correct path

WEEK-5

AIM: Create a calculator using the operations including addition, subtraction Multiplication and division using multilevel inheritance and display the desired Output

Class Diagram:



Code:

```
class addition
{
  public int add(int a, int b)
```

```
  {
      int addition = a+b;
      return addition;
  }
}
class subtraction extends addition
{
    public int sub(int a, int b)
    {
        int subtraction = a-b;
        return subtraction;
    }
}
class multiplication extends subtraction
{
     public int mult(int a, int b)
    {
        int multiplication = a*b;
        return multiplication;
    }
}
class division extends multiplication
{
   public int div(int a,int b)
   {
```

```
        int division = a/b;
        return division;
    }
}
class calculator
{
    public static void main(String args[])
    {
        division obj = new division();
       System.out.println("Addition is:"+ obj.add(10,2));
        System.out.println ("Subtraction is:"+obj.sub(8,4));
         System.out.println("Multiplication is:"+obj.mult(12,4));
         System.out.println("Division is:"+obj.div(8,4));

    }
}
```

Output

```
C:\Users\tejab>cd C:\java files\Lab files java

C:\java files\Lab files java>javac calculator.java

C:\java files\Lab files java>java calculator
Addition is:12
Subtraction is:4
Multiplication is:48
Division is:2

C:\java files\Lab files java>
```

Error Table

S.No	Error Type	Cause	Rectification
1	Constructor error	Invalid name to method	Defined class name
2	Syntax error	Expected ‘(‘	Added parenthesis
3	Logical error	Incorrect arithmetic operation	Correct operation rectified

Negative Case:

```
C:\Users\tejab>cd C:\java files\Lab files java

C:\java files\Lab files java>javac calculator.java
calculator.java:44: error: reached end of file while parsing
}
 ^
1 error

C:\java files\Lab files java>
```

Important Points

Inheritence:

The concept of OOP where a class inherits the properties and behaviours from

Another class (parent class) which promotes code reusability and hieratchical relationships

Multilevel Inheritence:

This is a type of inheritance in which a class inherited from another class, and

That superclass, in turn, inherits from yet another class, creating a chain of

inheritence

extends:

The extends keyword defines the relation of child class with the parent class

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, bikes and truck and they need a program to

Store details about each vehicle, such as brand and speed

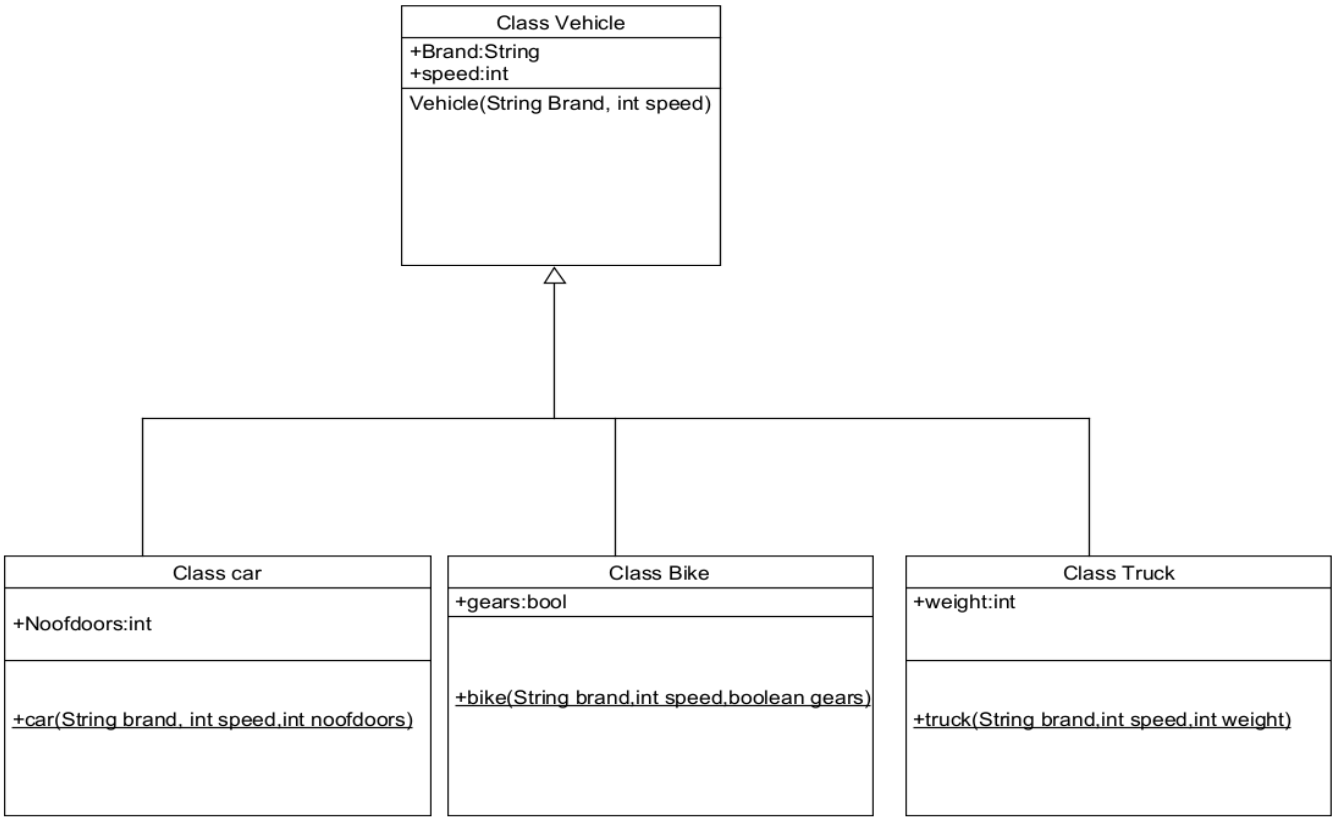
Cars should have an additional property: number of doors

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

Class diagram



Code:

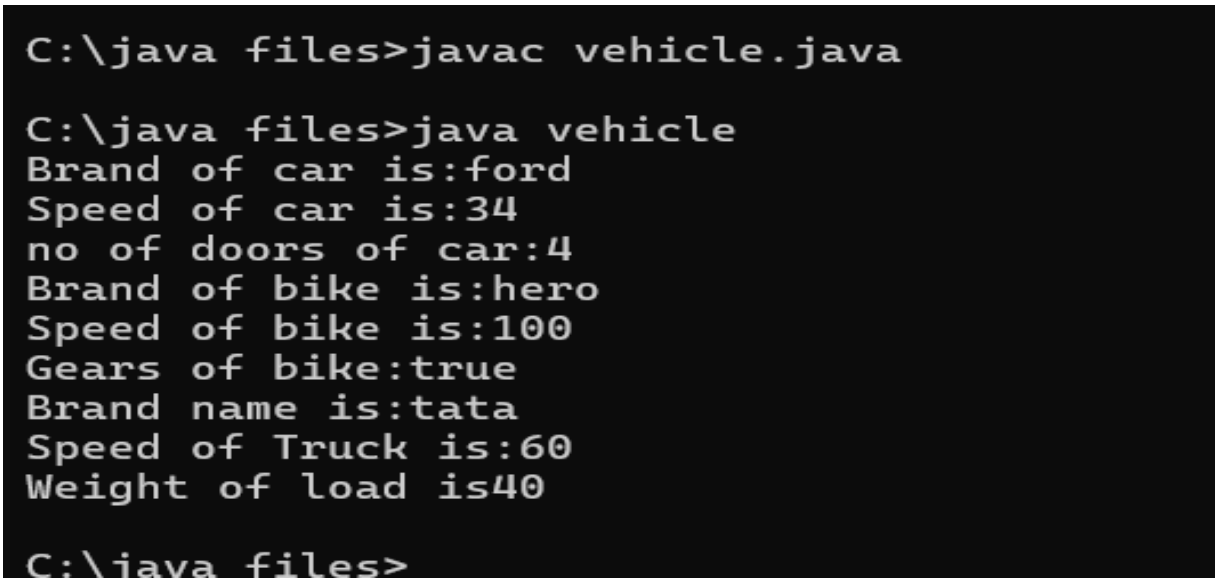
class vehicle{

```
    String brand;
    int speed;

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

    public static void main(String[] args) {

        car obj1=new car("ford",34,4);
        bike obj2=new bike("hero",100,true);
        truck obj3=new truck("tata",60,40);

    }
}

class car extends vehicle{

    int noofdoors;

    public car(String brand, int speed,int noofdoors) {
        super(brand, speed);
        this.noofdoors=noofdoors;
        System.out.println("Brand of car is:"+brand);
```

```
        System.out.println("Speed of car is:"+speed);
        System.out.println("no of doors of car:"+noofdoors);
    }
}

class bike extends vehicle{
    boolean gears;
 public bike(String brand,int speed,boolean gears){

        super(brand, speed);
        this.gears=gears;
        System.out.println("Brand of bike is:"+brand);
        System.out.println("Speed of bike is:"+speed);
        System.out.println("Gears of bike:"+gears);
    }
}

class truck extends vehicle{
    int weight;
    public truck(String brand,int speed,int weight){
        super(brand,speed);
        this.weight=weight;
        System.out.println("Brand name is:"+brand);
        System.out.println("Speed of Truck is:"+speed);
        System.out.println("Weight of load is"+weight);
```

```
  }
}
```

Output:



Error Table:

S No	Error Type	Cause	Rectification
1	Syntax Error	Semicolon missing	Added ;
2			
3			

Negative Case:

```
C:\java files\Lab files java>cd C:\java files

C:\java files>javac vehicle.java
vehicle.java:45: error: invalid method declaration; return type require
    public truck(String brand,int speed,int weight){
           ^
1 error

C:\java files>
```

Important Points

Hierarchical Inheritence:

This is a type of inheritance occurs when multiple subclasses inherit from a

Single parent class

v

WEEK-6

1) **Write a Java program to create a vehicle class with a method displayInfo(). Override this method in the car subclass to provide specific information about a car, model, fuel type, and color using the constructor**

- Write your code in VS CODE and execute

- Important Points:

0. Understand the calling of a Constructor
1. Giving class name correctly
2. Give the parameters Correctly

CODE:

```
class Vehicle {
   String Brand;
   String model;
   String fuel;
   String color;
   int capacity;

   Vehicle(String Brand, String model, String fuel, int capacity, String color) {
      this.Brand = Brand;
      this.model = model;
      this.fuel = fuel;
      this.capacity = capacity;
      this.color = color;
   }

   void displayInfo(String Brand, String model, String fuel, int capacity, String color) {
      System.out.println("Vehicle Details: ");
      System.out.println("Brand: " + Brand);
```

```
        System.out.println("Model: " + model);
        System.out.println("Fuel: " + fuel);
        System.out.println("Capacity: " + capacity);
        System.out.println("Color: " + color);
    }
}

class Car extends Vehicle {

    Car(String Brand, String model, String fuel, int capacity, String color) {
        super(Brand, model, fuel, capacity, color);
    }
    void displayInfo() {
        System.out.println("Car Details: ");
        System.out.println("Brand: " + Brand);
        System.out.println("Model: " + model);
        System.out.println("Fuel: " + fuel);
        System.out.println("Capacity: " + capacity);
        System.out.println("Color: " + color);
    }
}
class Week6_1 {
    public static void main(String[] args) {
        // Creating an instance of Car
        Car car1 = new Car("BMW", "X5", "Petrol", 6, "Red");
```

```
        car1.displayInfo();  // Display car details
    }
}
```

OUTPUT:

```
Car Details:
Brand: BMW
Model: X5
Fuel: Petrol
Capacity: 6
Color: Red
```

Errors:

S.NO	Error Name	Error Rectification
1	Syntax/ Compilation Error	Absence of Semicolon
2	Closing Brackets	Need to Close the brackets
3	Class Name Error	Give the class name correctly
4	Constructor Calling	Call the constructor correctly

2) Create a Java program for the scenario.

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

i) UG admissions require a minimum of 60%

ii) PG admissions require a minimum of 70%

- Write your code in VS CODE and execute

- Important Points:

3. Understand the calling of a Constructor

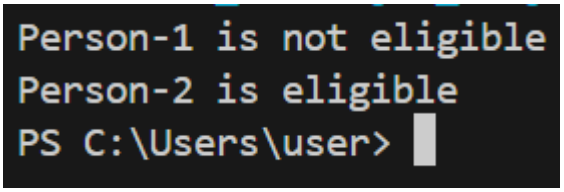
4. Giving class name correctly
5. Give the parameters Correctly

CODE:

```
class College{
   String name;
   int percentage;
   void geteligibility(String name,int percentage){
      this.name=name;
      this.percentage=percentage;
   }
}
class UG extends College{
   void geteligibility(String name,int percentage){
      if (percentage>=60){
         System.out.println(name+" is eligible");
      }
      else{
         System.out.println(name+" is not eligible");
      }
   }
}
class PG extends College{
```

```
    void geteligibility(String name,int percentage){
        if (percentage>=70){
            System.out.println(name+" is eligible");
        }
        else{
            System.out.println(name+" is not eligible");
        }
    }
}
class week6_2{
    public static void main(String[] args){
        UG ug=new UG();
        ug.geteligibility("Person-1",40);
        PG pg=new PG();
        pg.geteligibility("Person-2",80);
    }
}
```

OUTPUT:



Errors:

S.NO	Error Name	Error Rectification
1	Syntax/ Compilation Error	Absence of Semicolon
2	Closing Brackets	Need to Close the brackets

3	Class Name Error	Give the class name correctly
4	Constructor Calling	Call the constructor correctly

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

i) Add two integers

ii) Add two doubles

iii) Add three integers

- Write your code in VS CODE and execute

- Important Points:

6. Understand the calling of a Constructor
7. Giving class name correctly
8. Give the parameters Correctly

CODE:

```
class Calc{

   public int add(int a,int b){
```

```
        return a+b;
    }
    public double add(double a,double b){
        return a+b;
    }
    public int add(int a,int b,int c){
        return a+b+c;
    }
}
class week6_3{
    public static void main(String[] args){
        Calc C1=new Calc();
        System.out.println("Sum of 6 and 9 is: "+C1.add(6,9));
        System.out.println("Sum of 7.6 and 8.6 is: "+C1.add(7.6,8.6));
        System.out.println("Sum of 2,4 and 6 is: "+C1.add(2,4,6));
    }
}
```

OUTPUT:

```
Sum of 6 and 9 is: 15
Sum of 7.6 and 8.6 is: 16.2
Sum of 2,4 and 6 is: 12
PS C:\Users\user>
```

Errors:

S.NO	Error Name	Error Rectification
1	Syntax/ Compilation Error	Absence of Semicolon
2	Closing Brackets	Need to Close the brackets
3	Class Name Error	Give the class name correctly
4	Constructor Calling	Call the constructor correctly

4) Write a Java Program to create a shape class with a method calculateArea() that is overloaded for different shapes(e.g., Square, Rectangle). Then create a subclass Circle that overrides the calculateArea() method for a circle.

- Write your code in VS CODE and execute

- Important Points:

9. Understand the calling of a Constructor
10. Giving class name correctly
11. Give the parameters Correctly

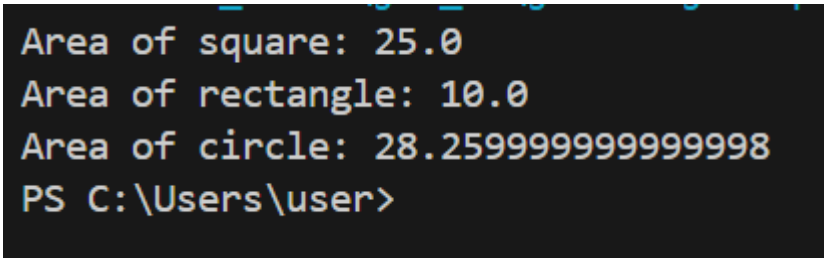
CODE:

```
class Shape {

   double calculateArea(double side) {
```

```
        return side * side;
    }

    double calculateArea(double width, double height) {
        return width * height;
    }
}

class Circle extends Shape {

    double calculateArea(double radius) {
        return 3.14 * radius * radius;
    }
}

class Week6_4 {
    public static void main(String[] args) {
        Shape S1 = new Shape();


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


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

```
        Circle C1 = new Circle();



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

OUTPUT:



Errors:

S.NO	Error Name	Error Rectification
1	Syntax/ Compilation Error	Absence of Semicolon
2	Closing Brackets	Need to Close the brackets
3	Class Name Error	Give the class name correctly
4	Constructor Calling	Call the constructor correctly

WEEK-6

1) **Write a Java program to create a vehicle class with a method displayInfo(). Override this method in the car subclass to provide specific information about a car, model, fuel type, and color using the constructor**

- Write your code in VS CODE and execute

- Important Points:

1. Understand the calling of a Constructor
2. Giving class name correctly
3. Give the parameters Correctly

CODE:

```
class Vehicle {
        String Brand;
        String model;
        String fuel;
        String color;
        int capacity;

        Vehicle(String Brand, String model, String fuel, int capacity, String color) {
           this.Brand = Brand;
           this.model = model;
           this.fuel = fuel;
           this.capacity = capacity;
           this.color = color;
        }

        void displayInfo(String Brand, String model, String fuel, int capacity, String
color) {
           System.out.println("Vehicle Details: ");
           System.out.println("Brand: " + Brand);
           System.out.println("Model: " + model);
```

```
        System.out.println("Fuel: " + fuel);
        System.out.println("Capacity: " + capacity);
        System.out.println("Color: " + color);
    }
}

class Car extends Vehicle {

    Car(String Brand, String model, String fuel, int capacity, String color) {
        super(Brand, model, fuel, capacity, color);
    }
    void displayInfo() {
        System.out.println("Car Details: ");
        System.out.println("Brand: " + Brand);
        System.out.println("Model: " + model);
        System.out.println("Fuel: " + fuel);
        System.out.println("Capacity: " + capacity);
        System.out.println("Color: " + color);
    }
}
```

```
class Week6_1 {
    public static void main(String[] args) {
        // Creating an instance of Car
        Car car1 = new Car("BMW", "X5", "Petrol", 6, "Red");
        car1.displayInfo();  // Display car details
    }
}
```

OUTPUT:

```
Car Details:
Brand: BMW
Model: X5
Fuel: Petrol
Capacity: 6
Color: Red
```

Errors:

S.NO	Error Name	Error Rectification
1	Syntax/ Compilation Error	Absence of Semicolon
2	Closing Brackets	Need to Close the brackets
3	Class Name Error	Give the class name correctly

4	Constructor Calling	Call the constructor correctly

2) Create a Java program for the scenario.

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

i) UG admissions require a minimum of 60%

ii) PG admissions require a minimum of 70%

- Write your code in VS CODE and execute

- Important Points:

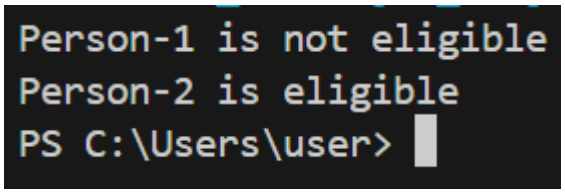
4. Understand the calling of a Constructor
5. Giving class name correctly
6. Give the parameters Correctly

CODE:

```
class College{
   String name;
   int percentage;
   void geteligibility(String name,int percentage){
      this.name=name;
      this.percentage=percentage;
   }
}
class UG extends College{
   void geteligibility(String name,int percentage){
      if (percentage>=60){
         System.out.println(name+" is eligible");
      }
      else{
         System.out.println(name+" is not eligible");
      }
   }
}
class PG extends College{
   void geteligibility(String name,int percentage){
```

```
        if (percentage>=70){
            System.out.println(name+" is eligible");
        }
        else{
            System.out.println(name+" is not eligible");
        }
    }
}
class week6_2{
    public static void main(String[] args){
        UG ug=new UG();
        ug.geteligibility("Person-1",40);
        PG pg=new PG();
        pg.geteligibility("Person-2",80);
    }
}
```

OUTPUT:



Errors:

S.NO	Error Name	Error Rectification
1	Syntax/ Compilation Error	Absence of Semicolon
2	Closing Brackets	Need to Close the brackets
3	Class Name Error	Give the class name correctly
4	Constructor Calling	Call the constructor correctly

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

i) Add two integers

ii) Add two doubles

iii) Add three integers

- Write your code in VS CODE and execute

- Important Points:

7. Understand the calling of a Constructor
8. Giving class name correctly
9. Give the parameters Correctly

CODE:

```
class Calc{

   public int add(int a,int b){
      return a+b;
   }
   public double add(double a,double b){
      return a+b;
   }
   public int add(int a,int b,int c){
      return a+b+c;
```

```
    }
}
class week6_3{
    public static void main(String[] args){
        Calc C1=new Calc();
        System.out.println("Sum of 6 and 9 is: "+C1.add(6,9));
        System.out.println("Sum of 7.6 and 8.6 is: "+C1.add(7.6,8.6));
        System.out.println("Sum of 2,4 and 6 is: "+C1.add(2,4,6));
    }
}
```

OUTPUT:

```
Sum of 6 and 9 is: 15
Sum of 7.6 and 8.6 is: 16.2
Sum of 2,4 and 6 is: 12
PS C:\Users\user>
```

Errors:

S.NO	Error Name	Error Rectification
1	Syntax/ Compilation Error	Absence of Semicolon

2	Closing Brackets	Need to Close the brackets
3	Class Name Error	Give the class name correctly
4	Constructor Calling	Call the constructor correctly

4) Write a Java Program to create a shape class with a method calculateArea() that is overloaded for different shapes(e.g., Square, Rectangle). Then create a subclass Circle that overrides the calculateArea() method for a circle.

- Write your code in VS CODE and execute

- Important Points:

10. Understand the calling of a Constructor

11. Giving class name correctly
12. Give the parameters Correctly

CODE:

```
class Shape {

    double calculateArea(double side) {
        return side * side;
    }

    double calculateArea(double width, double height) {
        return width * height;
    }
}

class Circle extends Shape {

    double calculateArea(double radius) {
        return 3.14 * radius * radius;
    }
}
```

```
class Week6_4 {
    public static void main(String[] args) {
        Shape S1 = new Shape();



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



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



        Circle C1 = new Circle();



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

OUTPUT:

```
Area of square: 25.0
Area of rectangle: 10.0
Area of circle: 28.259999999999998
PS C:\Users\user>
```

Errors:

S.NO	Error Name	Error Rectification
1	Syntax/ Compilation Error	Absence of Semicolon
2	Closing Brackets	Need to Close the brackets
3	Class Name Error	Give the class name correctly
4	Constructor Calling	Call the constructor correctly