\\

LAB MANUAL

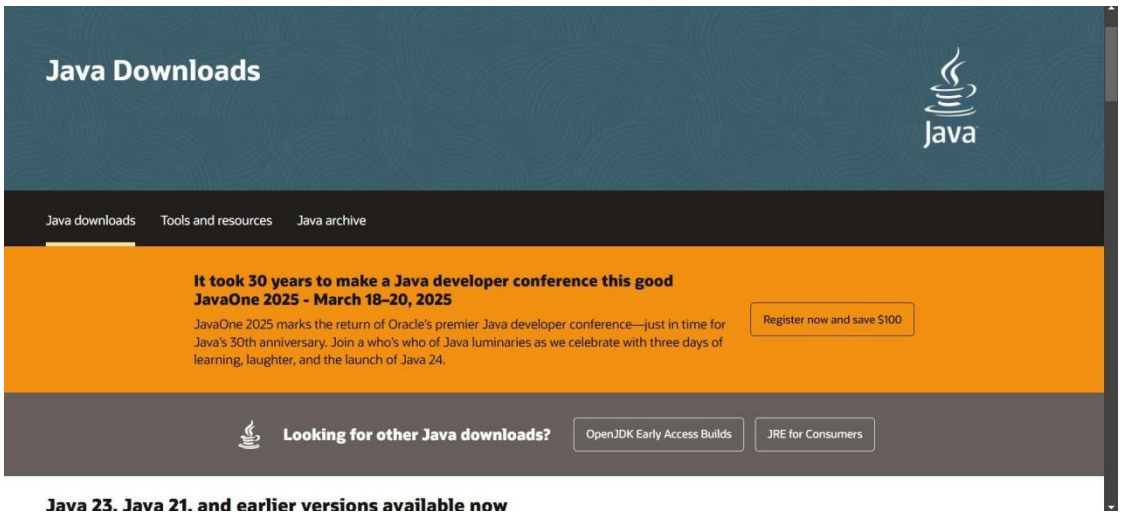
ROLLNO:AV.SC.U4CSE24132
NAME: j.bhagiradha
SECTION: CSE-B

\\

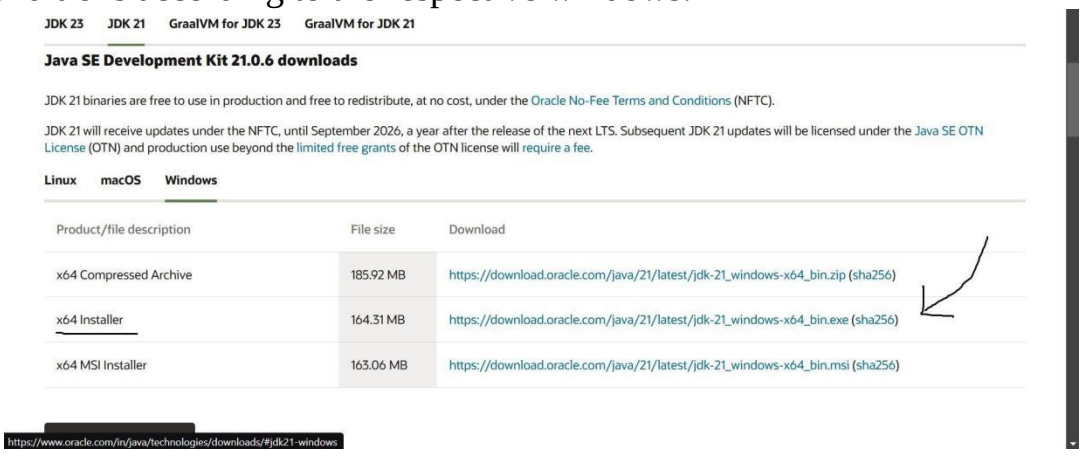
WEEK-1:

Aim: How to install jdk and first program on printing student details.

Step-1: Download JDK-21 from oracle website

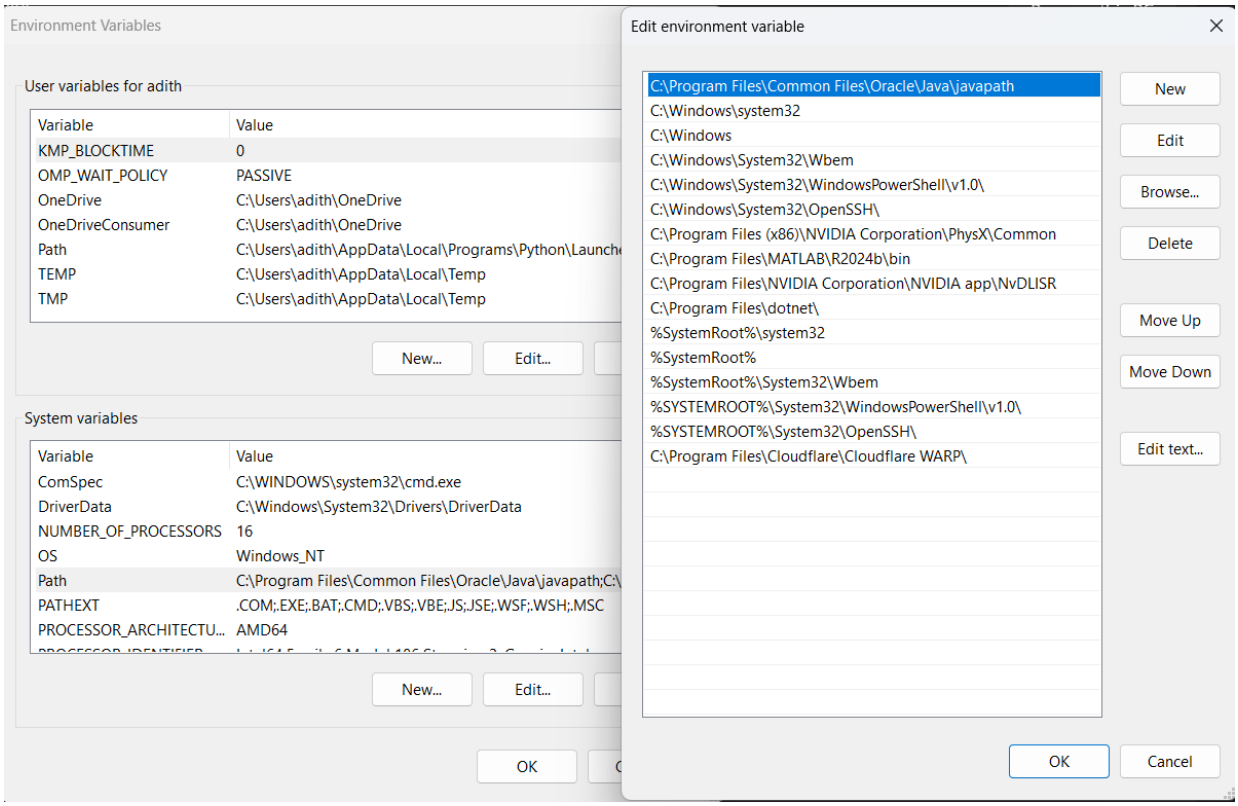


Step-2: Install the JDK-21 with accepting terms and conditions according to the respective windows.



\\

Step-3:Setting up environmental variables.



*Windows c -> C-drive -> program files ->Java ->JDK-21->select bin

*Select and open environmental variable in search bar-> either select system variables or user variables-> select path-> click edit->New-> paste the bin-> finish the setup(apply the changes).

~for verifying the installed version

Open cmd-> type java --version

~command propt

Javac filename.java ->compiling.

Java filename.java ->displaying

\\

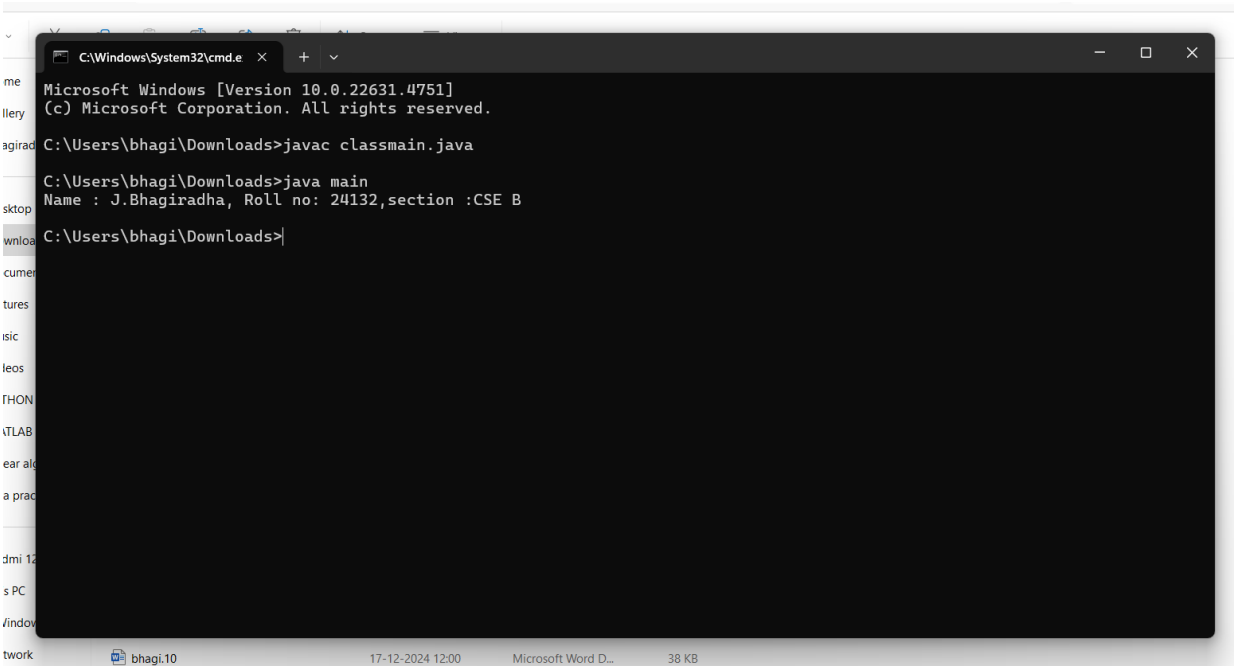
PROGRAM-1(Rectified):

swapping.rap | untitled5.m | filename3.html | Main.java | bhagisimportant

File Edit View

```
class  main{
public static void main(String[ ]args){
 System.out.println("Name : J.Bhagiradha, Roll no: 24132,section :CSE B");
   }
}
```

Output:



\\

WEEK-2:

PROGRAM-1:

Aim: Write a java program for SI

```
File    Edit    View

import java.util.Scanner;

        public class si{
        public static void main(String[] args)
        { int p,t,r, Simpleinterst;
        Scanner num=new Scanner(System.in);
        System.out.println("enter the value of p");
        p=num.nextInt();
        System.out.println("Enter the value of t");
        t=num.nextInt();
        System.out.println("enter the value of r");
        r=num.nextInt();
        Simpleinterst=(p*t*r)/100;
        System.out.println("Simpleinterst: " + Simpleinterst);
}
}
```

Output:

```
C:\Windows\System32\cmd.e

Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bhagi\OneDrive\Desktop\java labs>javac javar.java

C:\Users\bhagi\OneDrive\Desktop\java labs>java javar.java
Enter the princpial value10
Enter the rate of interest value20
Enter the time value2
4.0
C:\Users\bhagi\OneDrive\Desktop\java labs>
```

ERROR TABLE:

Code Error	Code rectification

\\

1.Giving space between next and Double. 2.Not giving parenthesis after closing the input.	1.Should not give space between next and Double. 2.We must put parenthesis after closing the input.

\\

PROGRAM-2 :
Aim : Write a program in java for area of rectangle.

```
File    Edit    View

import java.util.Scanner;

        public class areaofrectangle {
        public static void main(String[] args) {
        int l, b, Areaofrectangle;
        Scanner measurement = new Scanner(System.in);
        System.out.println("enter the value of length");
        l = measurement.nextInt();
        System.out.println("Enter the value of breadth");
        b = measurement.nextInt();
        Areaofrectangle =l*b;
        System.out.println("Areaofrectangle:" + Areaofrectangle);
 }
}
```

Output:

```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bhagi\OneDrive\Desktop\java labs>javac rectangle.java

C:\Users\bhagi\OneDrive\Desktop\java labs>java rectangle.java
Enter the length : 10
Enter the breath : 20
200
C:\Users\bhagi\OneDrive\Desktop\java labs>
```

ERROR TABLE:

Code Error	**Code rectification**
1.While using for iteration, not giving the conditions correctly. 2.Declaring the data type as double instead of int.	1.We should give iterative statements correctly. 2.We should give the data type as int for integers.

\\

\\

PROGRAM-3:

Aim: Write a program in java for area of triangle using heron's formula.

File Edit View

```
import java.util.Scanner;

        public class triangle {
        public static void main(String[] args){
        Scanner input=new Scanner(System.in);
        System.out.println("a: ");
        double a=input.nextDouble();
        System.out.println("b: ");
        double b=input.nextDouble();
        System.out.println("c: ");
        double c=input.nextDouble();
        double s=a+b+c/2;
        double area = Math.sqrt(s*(s-a)*(s-b)*(s-c));
        System.out.println("Area:" + area);
        input.close();
}
}
```

Output:

```
PS A:\javaR> javac triangle.java
PS A:\javaR> java triangle.java
a:
18
b:
45
c:
2
Area:1862.2652872241376
PS A:\javaR>
```

ERROR TABLE:

Code Error	Code rectification
1.While printing the variable not giving + sign. 2.Not closing the scanner.	1.We should give correct indentation. 2.Closing the scanner is must.

\\

PROGRAM-4(a):

Aim: Write a program in java for converting temperature from celsius to fahrenite.

File Edit View

```
import java.util.Scanner;

        public class celsiustofarenheit {
        public static void main(String[] args) {
        int C,F;
        Scanner num = new Scanner(System.in);
        System.out.println("enter the value of C");
        C=num.nextInt();
        F=(C*9/5)+32;
        System.out.println("farenheit is:"+F);

}
}
```

C:\Windows\System32\cmd.e

```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bhagi\OneDrive\Desktop\java labs>javac ftoc.java

C:\Users\bhagi\OneDrive\Desktop\java labs>java ftoc.java
Enter the clesics value 69
the forenheat value is 101.0
C:\Users\bhagi\OneDrive\Desktop\java labs>
```

OUTPUT:

PROGRAM-4(b):

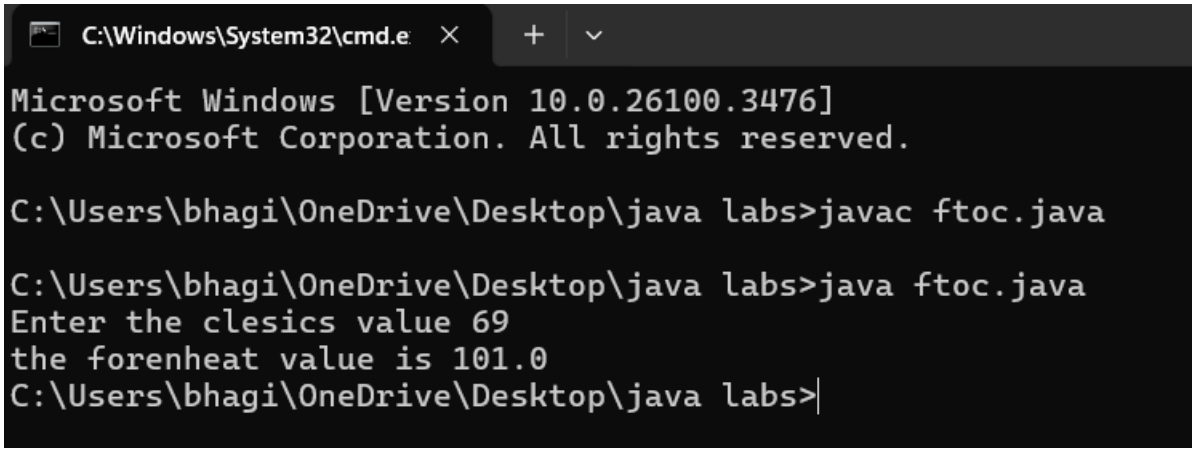
Aim: Write a program in java for converting temperature from fahrenite to celsius.

\\

File Edit View

```
import java.util.Scanner;

        public class farenheittocelsius {
        public static void main(String[] args) {
        int C,F;
        Scanner num = new Scanner(System.in);
        System.out.println("enter the value of F");
        F= num.nextInt();
        C=(F-32)*5/9;
        System.out.println("Celsius is: "+C);
}
}
```

Output:



ERROR TABLE:

Code Error	**Code rectification**
1.While printing the variable not giving + sign. 2.Not closing the scanner.	1.We should give correct indentation. 2.Closing the scanner is must.

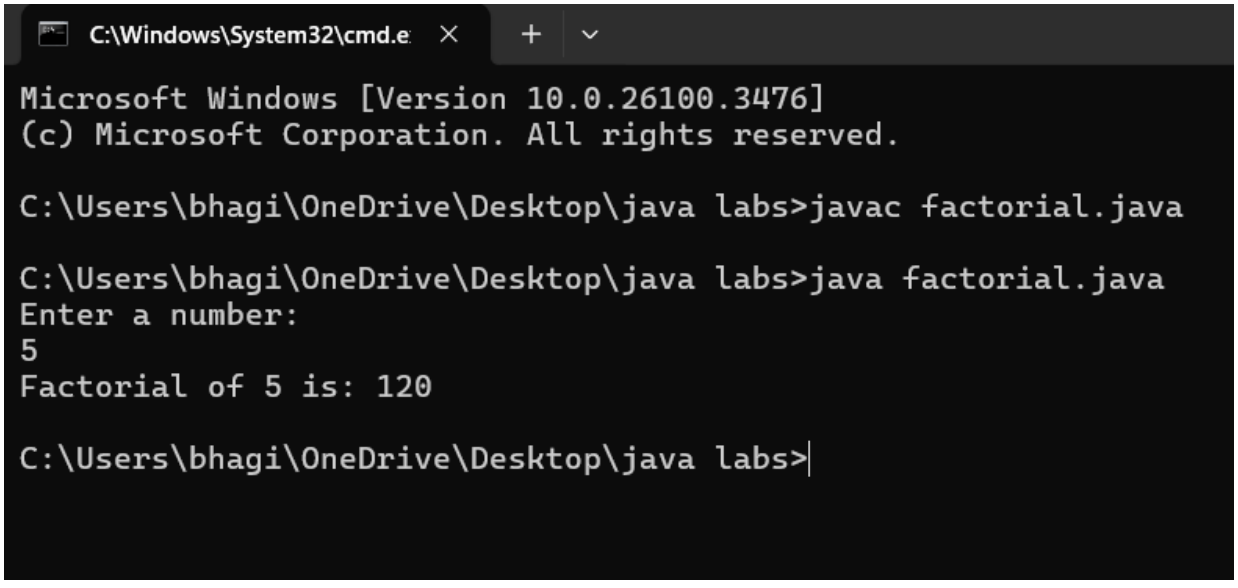
\\

PROGRAM-5:

Aim: Write a program in java for factorial of a number.

```
File    Edit    View

import java.util.Scanner;

        public class Factorial {
        public static void main(String[] args) {
        int n, factorial = 1;
        Scanner num = new Scanner(System.in);
        System.out.println("Enter a number:");
        n=num.nextInt();
        for (int i=1; i <= n; i++) {
        factorial*=i;
        }
        System.out.println("Factorial of "+n+" is: "+ factorial);
}
}
```

OUTPUT:



\\

ERROR TABLE:

Code Error	**Code rectification**
1.While using for iteration, not giving the conditions correctly. 2.Declaring the data type as double instead of int.	1.We should give iterative statements correctly. 2.We should give the data type as int for integers.

PROGRAM-6:

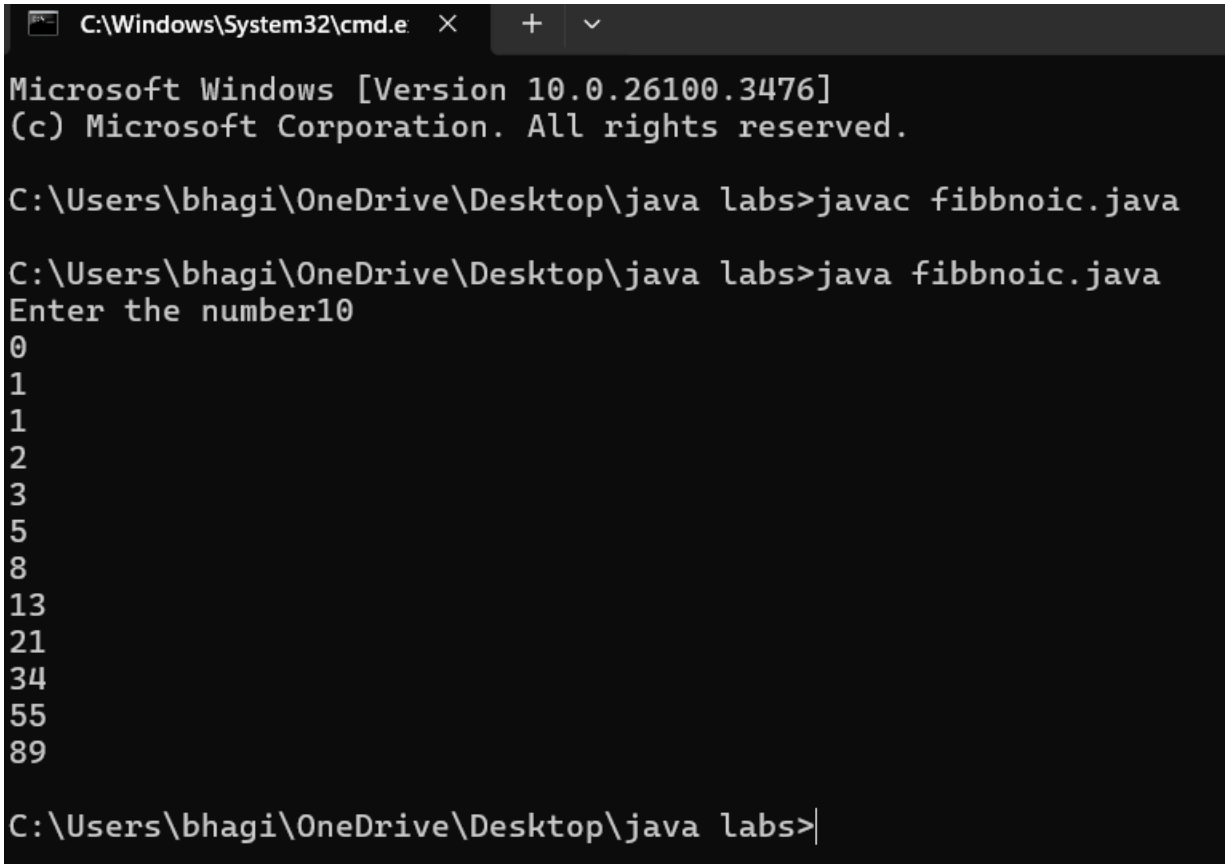
Aim: Write a program in java for fibonacci series.

File Edit View

```
import java.util.Scanner;

        public class fibonaccisequence {
        public static void main(String[]args){
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter the no of terms:");
        int n=scanner.nextInt();
        int a=0,b=1;
        System.out.println("Fibonacci sequence");
        for (int i=1;i<=n;i++){
        System.out.println(a+" ");
        int nextTerm =a+b; a=b;
        b=nextTerm;
        }
        scanner.close();
}
}
```

OUTPUT:

\\



ERROR TABLE:

Code Error	**Code rectification**
1.Giving space between next and Double. 2.Not giving parenthesis after closing the input.	1.Should not give space between next and Double. 2.We must put parenthesis after closing the input.

\\

WEEK -3:

PROGRAM-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 :

\\

```java
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:

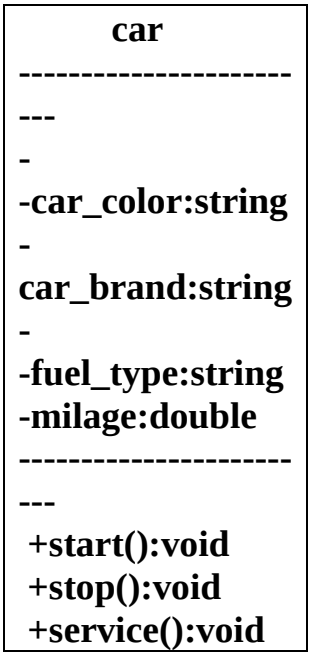
```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bhagi\OneDrive\Desktop\java labs>javac car.java

C:\Users\bhagi\OneDrive\Desktop\java labs>java car.java
car brand:Black car color:BMW car fuel-type:petrol car milage:150.0km
Black it is well known car BMW is iconic color petrol is source and of milage 150.0km
car brand:oddi car color:petrol car fuel-type:Blue car milage:130.0km
oddi it is well known car petrol is good color Blue is source and of milage 130.0km
car brand:Tesla car color:red car fuel-type:e-power car milage:200.0km
Tesla it is well known car for its speed red is beautiful colore-power is source and of milage 200.0km

C:\Users\bhagi\OneDrive\Desktop\java labs>
```

\\

Error table :

S.no	Error name	Cause of error	Rectification
1	Syntax Error	Missing ‘{‘	‘{‘ added
2	Compile time Error	Mispelled Variable call	Rectified with Correct variable name
3	Case sensitive error	Uppercase and lowercase	rectified

Class diagram:



PROGRAM-2:

Aim : To create a class BankAccount with methods deposit() and withdraw() create two subclasses savingsaccount and checkingaccount override the withdraw method in each subclass to impose different withdrawal limits and fees

\\

CODE :

```
import java.util.Scanner; class Bank_account{ long current_balance; String name;
String account_number;
String IFSE;
String branch;
Scanner input=new Scanner(System.in);
public Bank_account(long current_balance,String name,String account_number,String IFSE,String branch){
this.current_balance=current_balance; this.name=name; this.account_number=account_number; this.IFSE=IFSE;
this.branch=branch;
System.out.println("User name:"+name+" account_number:"+account_number+" IFSE details:"+IFSE+" branch number:"+branch);
}
public void deposit(){
System.out.println("enter the depositing amount: "); long deposit_amount=input.nextLong();
long sum=current_balance+deposit_amount; System.out.println("the current blance after depoisting is "+ sum);
}
public void withdraw(){
System.out.println("enter the withdrawing amount: "); long withdraw_amount=input.nextLong();
long dum=current_balance-withdraw_amount; if(dum>0){
System.out.println("the current blance after withdrawal is "+ dum);} else{
System.out.println("the current blance is insufficent ");
}
}
public static void main(String[] args){ Scanner input=new Scanner(System.in);
System.out.println("enter the Balance amount: "); long amount=input.nextLong();
Bank_account a1=new Bank_account(amount,"santhosh","99082761187","ascr23467","mangalgiri"); a1.deposit();
System.out.println("enter the Balance amount: "); long amount1=input.nextLong();
Bank_account a2=new Bank_account(amount1,"pandu","961456789543","asde345777","khammam"); a2.withdraw();
}
}
```

OUTPUT:

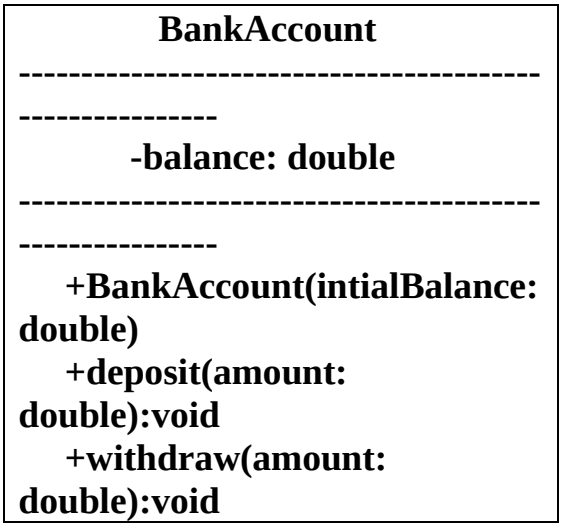
```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bhagi\OneDrive\Desktop\java labs>javac bank.java

C:\Users\bhagi\OneDrive\Desktop\java labs>java bank.java
enter the Balance amount:
500
User name:santhosh account_number:99082761187 IFSE details:ascr23467 branch number:mangalgiri
enter the depositing amount:
2000
the current blance after depoisting is 2500
enter the Balance amount:
500
User name:pandu account_number:961456789543 IFSE details:asde345777 branch number:khammam
enter the withdrawing amount:
50000
the current blance is insufficent

C:\Users\bhagi\OneDrive\Desktop\java labs>
```

\\

Error table:

S.no	Error name	Error name	Rectification
1	Name Error	Undefined name	Correct variable Name replaced
2	Syntax Error	Missing Parenthesis	Parenthesis Added
3	Logical Error	Incorrect Condition	Condition Rectified

Class diagram:



\\

Week 4:

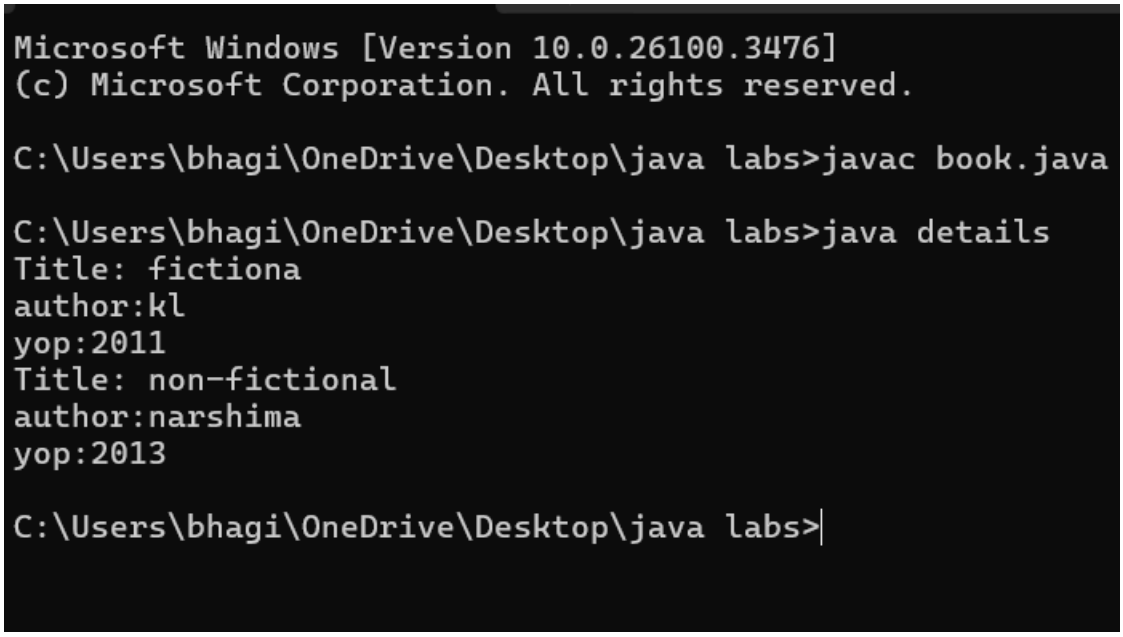
Program 1

Aim: Write a java program with class named ‘Book’ the class should contain various attributes such as title ,author, year of publish it should also contain a constructor with parameters which inisalizes title, author, year of publish .Create a method which display’s the details of the books ,Display the detailes of two books i.e create two objects and display their detailes.

```
class book{ String tittle; String author; int yop;
book(String tittle,String author,int yop){ this.tittle=tittle;
this.author=author; this.yop=yop;
}
void display(){ System.out.println("Title: "+tittle); System.out.println("author:"+author); System.out.println("yop:"+yop);
}
}
class details{
public static void main(String[] args) {
book b1=new book("fictiona","kl",2011); book b2=new book("non-fictional","narshima",2013); b1.display();
b2.display();
}
}
```

\\

\\
Output:



ERROR TABLE:

Code Error	**Code rectification**
1. Not defining the function in a file. 2. Two public class files should not be saved in the same file.	1. To call the method we must define a function in a file. 2. Two public class files should be saved in different files.

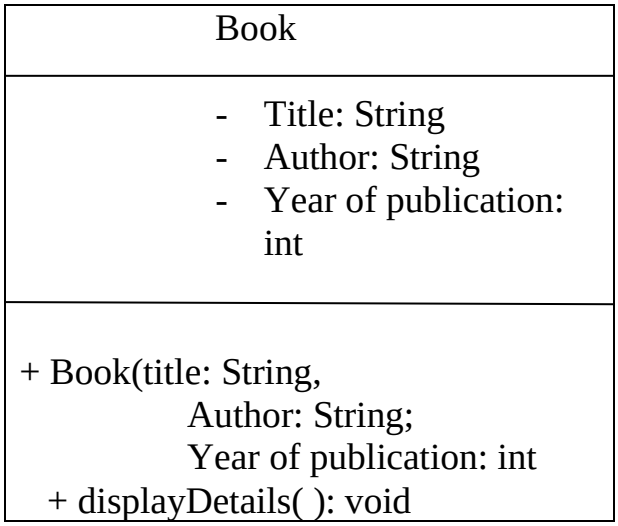
IMPORTANT POINTS:

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:



\\

Program 2:

Aim: Create a java program with class named “MyClass”, with a static variable count of int datatype and inisalized to zero and a constant variable ‘pi’
of type “double” insilaize to 3.145 as attributes of the class, Now define a count variable each time an object of “MyClass” is created.Finally print the final values of count and ‘pi’ variables.

Program:

```
class myclass{ static int count=0;
final double pi=3.1415; void myclass(){
count=count+1; System.out.println("Count: "+count); System.out.println("pi: "+pi);
}

}
class details{
public static void main(String[] args) { myclass obj1=new myclass(); myclass obj2=new myclass();



myclass obj3=new myclass(); myclass obj4=new myclass(); myclass obj5=new myclass(); obj1.myclass(); obj2.myclass(); obj3.myclass(); obj4.myclass(); obj5.myclass();

}
}
```

\\

```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bhagi\OneDrive\Desktop\java labs>javac pi.java

C:\Users\bhagi\OneDrive\Desktop\java labs>java details
Count: 1
pi: 3.1415
Count: 2
pi: 3.1415
Count: 3
pi: 3.1415
Count: 4
pi: 3.1415
Count: 5
pi: 3.1415

C:\Users\bhagi\OneDrive\Desktop\java labs>
```

ERROR TABLE:

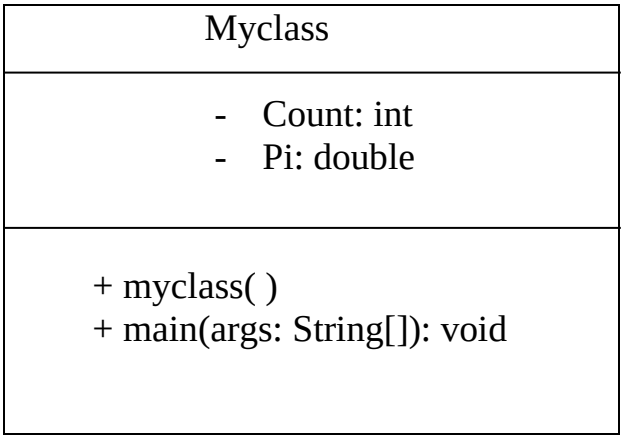
Code Error	**Code rectification**
1. Not Putting the semi-colon after calling a function, 2. Not giving the indentation properly.	1. Put the semi-colon after calling a function. 2. All the indentation must be correct to run the code correct.

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:

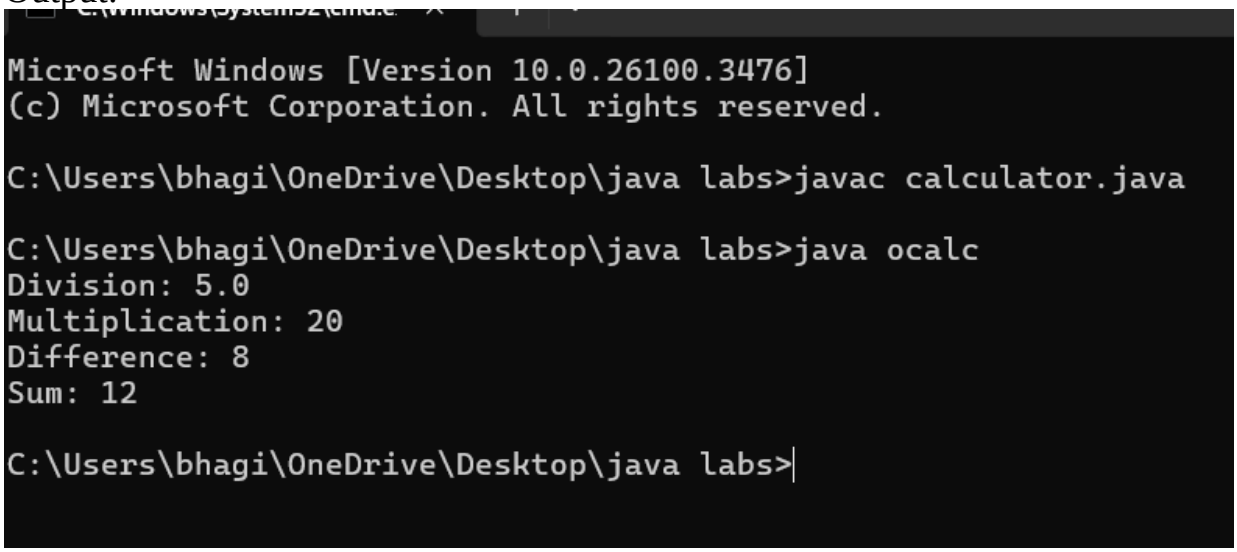
Aim: Create a calculator using the operations including addition, subtraction, multiplication and division using multilevel in heritance and display the desired output.

Program:

```
\\
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:



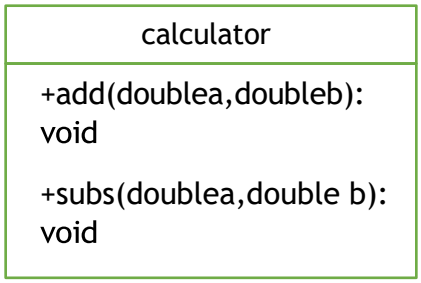
Errors:

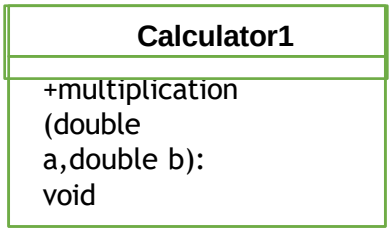
1.	**Semi colon (;)**	**Givethe semi colon (;) in each line where it is required**
2.	**Syntax Error**	**Giving Capital 'S' in printing statements (System.out.println)**

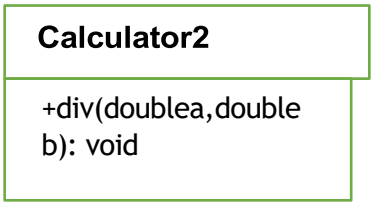
Sl.No	**Error name**	**Error Rectification**

\\

Class Diagram:







\\

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**.**

Questions:

1. Which OOPS concepts used in the above program? Explain why itis useful in this scenario.
2. If the company decides to add a new type of vehicle 'Truck', how would you modify the program?
3.

Truck should include and additionalpropertycapacity (in tons).

a. CreateashowTruck()methodtodisplaythetruck's capacity.
b. Write a constructor for truck that initializes all properties.
c. Implementthetruck classandupdatethemainmethodto createaTruckobjectand alsocreateanobjectforcarand bikesubclasses. Finallydisplay the details

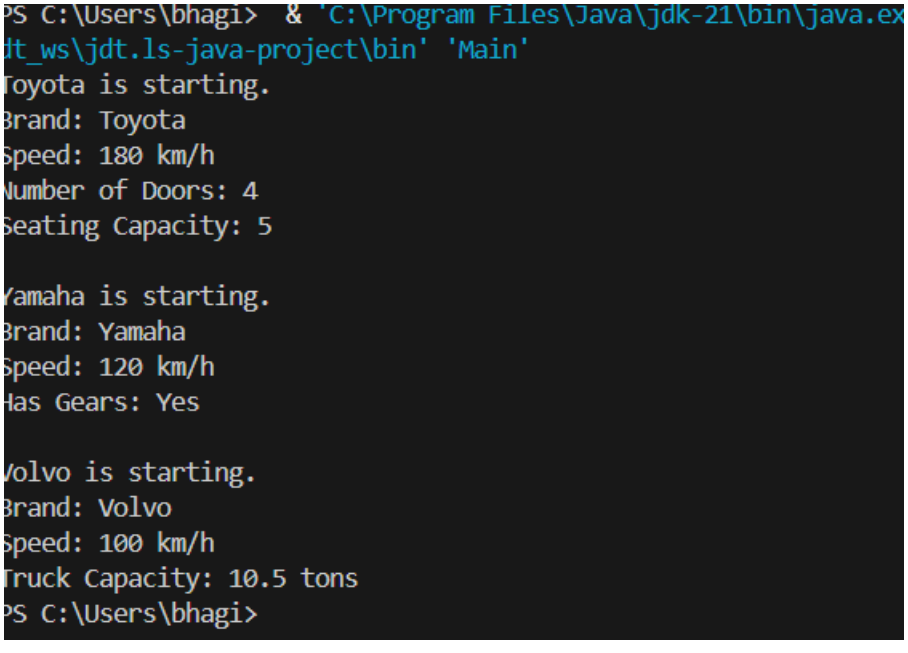
Program:

```
class Vehicle {
    protected String brand;
    protected 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 displayDetails() {
        System.out.println("Brand: " + brand);
        System.out.println("Speed: " + speed + " km/h");
    }
}

class Car extends Vehicle {
    private int numberOfDoors;
    private int seatingCapacity;

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

```
\\
        this.seatingCapacity = seatingCapacity;
    }

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


 }
}

class Bike extends Vehicle {
    private boolean hasGears;

    public Bike(String brand, int speed, boolean hasGears) {

  super(brand, speed);
        this.hasGears = hasGears;
    }

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

class Truck extends Vehicle {
    private double capacity; // in tons

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

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

    @Override
    public void displayDetails() {
```

```
\\
        super.displayDetails();
        showTruckDetails();
    }
}

 class Main {
    public static void main(String[] args) {
        // Create a Car object
        Car car = new Car("Toyota", 180, 4, 5);
        car.start();
        car.displayDetails();
        System.out.println();

        // Create a Bike object
        Bike bike = new Bike("Yamaha", 120, true);
        bike.start();
        bike.displayDetails();
        System.out.println();


        // Create a Truck object
        Truck truck = new Truck("Volvo", 100, 10.5);
        truck.start();
        truck.displayDetails();
    }
}
```

Output:



\\

Errors:

Sl.No	Error name	Error Rectification
1.	Semi colon (;)	Give the semi colon (;) in each line where it is required
2.	Syntax Error	Giving Capital ‘S’ in printing statements (System.out.println)

Class Diagram:

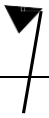
Vehicle
brand: string speed: string
+Vehicle(String brand, int speed) +Details(): void

CARS
doors: int

\\

capacity: int
+ CARS (String brand, int speed, int doors, int capacity) **+cardetails(): void**

Bikes
gears: Boolean
+ Bikes(String brand, int speed, Boolean gears) **+bikedetails(): void**



Trucks
tons: int
+ Trucks(String brand,int speed,int tons) **+truckdetails(): void**

Important points:

Multi-inheritance: It is one of the types of the inheritance where subclass 2 inherits subclass1 and subclass1 inherits superclass.

Here Vehicle is the super class or parent class and remaining cars, bikes, trucks are the subclasses or child classes

\\

Week 6:
Program 1:
Aim: Write a Java program to create a vehicle class with a method displayInfo(). Override this method in the car subclass to provide specific information about a car, model, fuel type, and colour using the constructor
Syntax:
Super class extends subclass
Here extends is the main key word which represents the extending relation from parent class to child class.

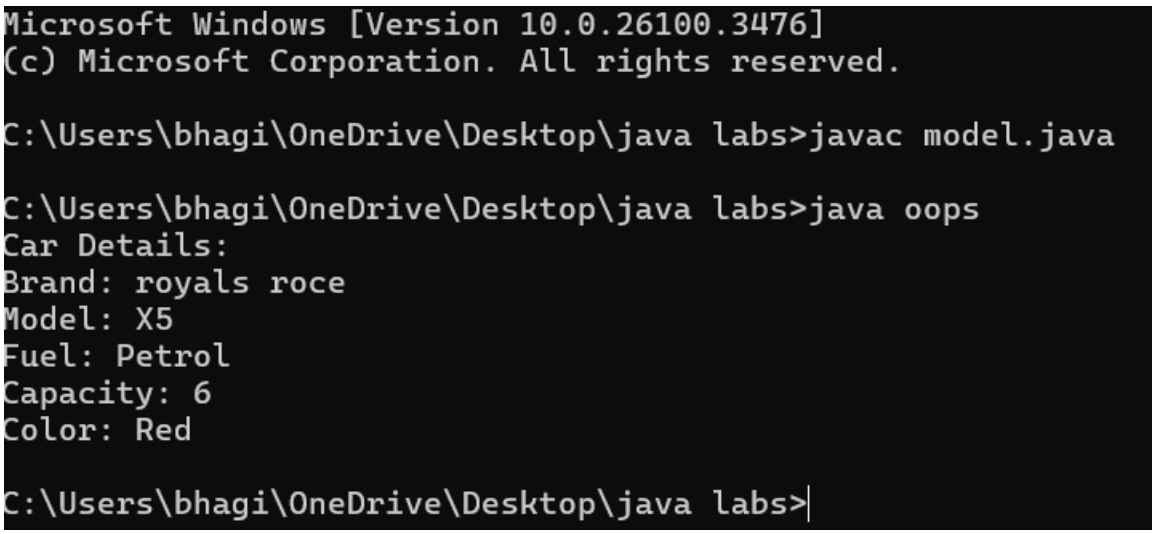
Program:

```
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 oops{
   public static void main(String[] args) {
   // Creating an instance of Car
```

\\

```
    Car car1 = new Car("royals roce", "X5", "Petrol", 6, "Red"); car1.displayInfo(); // x
    }
    }
```

Output:

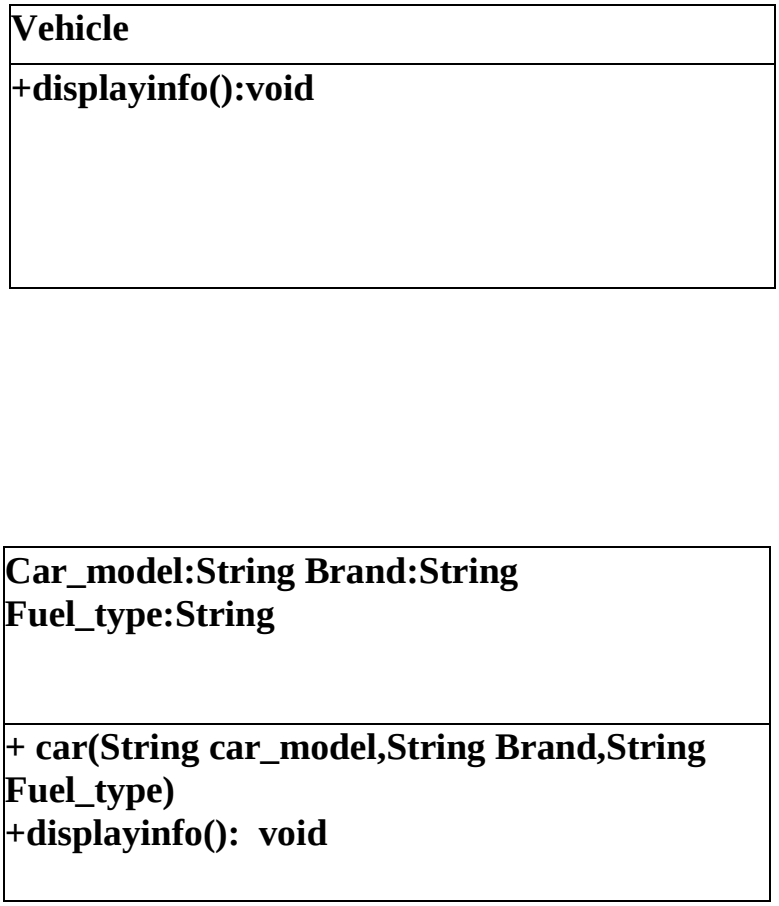


Errors:

Sl.No	Error name	Error Rectification
1.	Semi colon (;)	Give the semi colon (;) in each line where it is required
2.	Syntax Error	Giving Capital ‘S’ in printing statements (System.out.println)

\\

Class Diagram:



Important points:

In order to do this, we have to use inheritance concept. Here we used the multi-inheritance concept**.**

\\

Program 2:

Aim: Create a Java program for the scenario.

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

i) UG admissions require aminimum of 60%

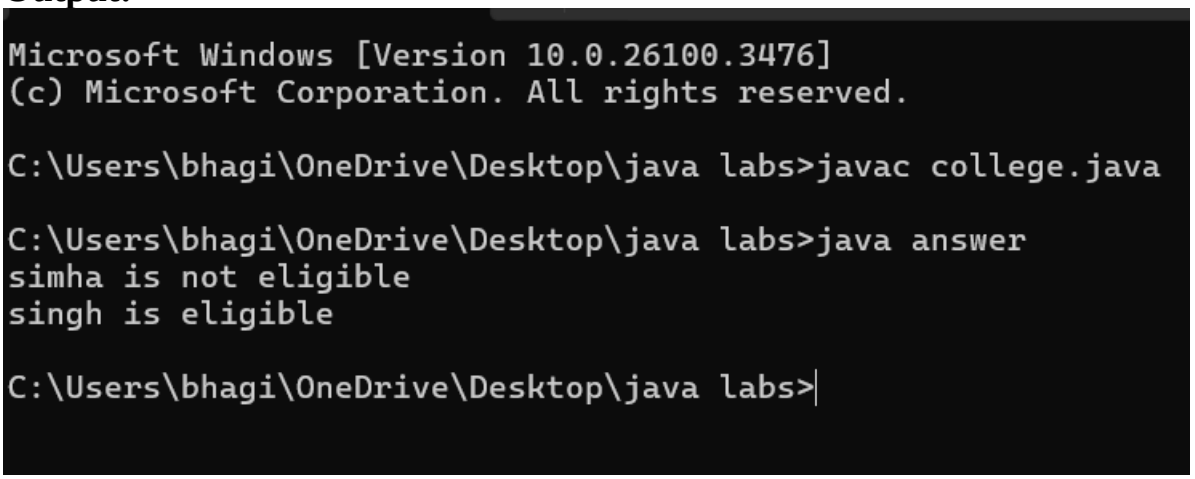
ii)PG admissions require aminimum of 70%

Program:

```
class College{ String name; int percentage;
void geteligibility(String name,int percentage){ this.name=name;
this.percentage=percentage;
}
}
class UG extends College{
void geteligibility(String name,int percentage){ if (percentage>=60){
System.out.println(name+" is eligible");


}
else{
System.out.println(name+" is not eligible");
}
}
}
class PG extends College{
void geteligibility(String name,int percentage){ if (percentage>=70){
System.out.println(name+" is eligible");
}
else{
System.out.println(name+" is not eligible");
}
}
}
```

\\
class answer{
public static void main(String[] args){ UG ug=new UG();
ug.geteligibility("simha",45); PG pg=new PG(); pg.geteligibility("singh",80);

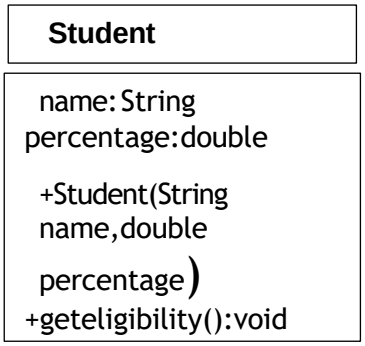
Output:

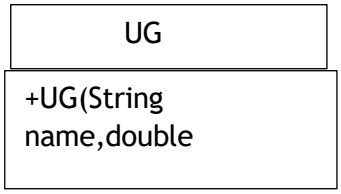


\\
Errors:

Sl.No	Error name	Error Rectification
1.	Semi colon (;)	Give the semi colon (;) in each line where it is required
2.	Syntax Error	Giving Capital 'S' in printing statements (System.out.println)

Important points:
Super keyword is used take the method,variable,constructor from the super class.
Class diagram:





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

i) Addtwointegers
ii) Addtwodoubles
iii) Addthreeintegers

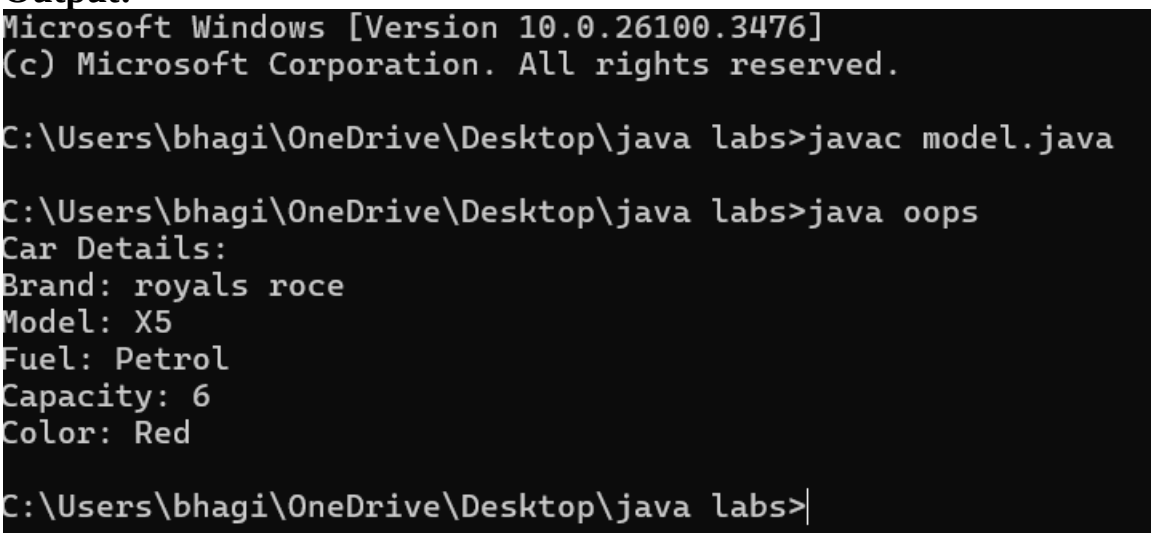
Important points:
We should carefully pass the double and integer and different types of input to an constructor when creating an object to access the different constructors based on the parameter.

\\

Program:

```
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 ayyayo{
public static void main(String[] args){ Calc C1=new Calc();
System.out.println("Sum of 24 and 10 is: "+C1.add(24,10)); System.out.println("Sum
of 8.9 and 7.5 is: "+C1.add(7.6,8.6)); System.out.println("Sum of 2,4 and 6 is:
"+C1.add(2,4,6));
}
}
```

Output:



Class diagram:

Calculatoroverloading
+ add(int a,int b):int **+add(double a,double b):double** **+ add(int a,int b,int c):int**

Errors:

Sl.No	Error name	Error Rectification
1. \\	Semi colon (;)	Give the semi colon (;) in each line where it is required
2.	Syntax Error	Giving Capital ‘S’ in printing statements (System.out.println)

Program 4:

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

Important points:

In this program we use both method overloading and overriding to calculate area of different shapes.

Class Diagram:

shape
+calculatearea(int l,intb):void +calculatearea(int x):void

circle
+void calculatearea(double pi,double r):void

\\

PROGRAM:

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

System.out.println("Area of square: " + S1.calculateArea(3));
System.out.println("Area of rectangle: " + S1.calculateArea(5, 2)); Circle C1 = new
Circle();
System.out.println("Area of circle: " + C1.calculateArea(5));
}
}
```

\\

Output:

```
C:\Users\bhagi\OneDrive\Desktop\java labs>javac shape.java

C:\Users\bhagi\OneDrive\Desktop\java labs>java ghible
Area of square: 9.0
Area of rectangle: 10.0
Area of circle: 78.5

C:\Users\bhagi\OneDrive\Desktop\java labs>
```

Errors:

Sl.No	Error name	Error Rectification
1.	Semi colon (;)	Give the semi colon (;) in each line where it is required
2.	Syntax Error	Giving Capital 'S' in printing statements (System.out.println)

\\

WEEK-7

1) **Aim : Write a Java program to create an abstract class Animal with an abstract method called sound(). Create subclasses Lion and Tiger that extend the Animal class and implement the sound() method to make a specific sound for each animal.**

Important points :

1. Here we used abstract class concept it is a restricted class that cannot be instantiated (cannot have objects created directly) and is typically designed to be extended by subclasses.

2. An abstract method is a method declared in an abstract class that does not have an implementation, meaning it doesn't have a body within the abstract class.

Program :

```
abstract  class Animal {
   abstract void Sound();

}
class lion extends Animal {
   public void Sound(){
      System.out.println("Roar");

   }

}
class Tiger extends Animal {
   public void Sound(){
      System.out.println("heehee");

   }

}
class main {
   public static void main(String[] args){
System.out.println("NAME:j.bhagiradha ,ROLL NO
```

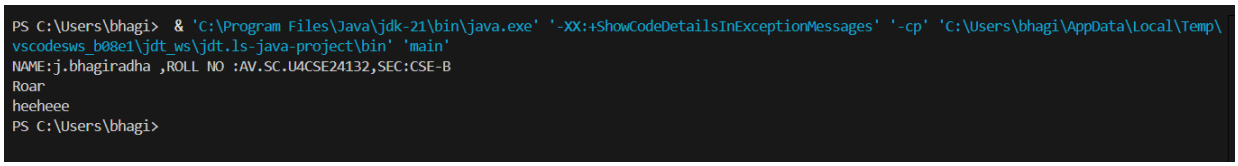
\\

:AV.SC.U4CSE24132,SEC:CSE-B");

S.NO	Error Name	Error Rectification
1	Main Class	Better to create main class name same as the file you saved and first letter is capital.
2	Method	We need to provide return type to the method.
3	Data type	As per need provide data type
4	Abstract method	Implementation in subclass only

```
lion l =new lion();
l.Sound();
Tiger t = new Tiger();
t.Sound();
    }}
```

Output:



Error table :

S.NO	Error Name	Error Rectification
1	Main Class	Better to create main class name same as the file you saved and first letter is capital.
2	Method	We need to provide return type to the method.
3	Data type	As per need provide data type
4	Abstract method	Implementation in subclass only

\\

2) **Aim : Write a Java program to create an abstract class Shape3D with abstract methods calculateVolume() and calculateSurfaceArea(). Create subclasses Sphere and Cube that extend the Shape3D class and implement the respective methods to calculate the volume and surface area of each shape.**

Important points :

1.Here we used abstract class concept it is a restricted class that cannot be instantiated (cannot have objects created directly) and is typically designed to be extended by subclasses.

2.An abstract method is a method declared in an abstract class that does not have an implementation, meaning it doesn't have a body within the abstract class.

Program :

```
abstract class Shape3D {
    abstract double calculateVolume();
    abstract double calculateSurfaceArea();
}

class Sphere extends Shape3D {
    private double radius;

    Sphere(double radius) {
        this.radius = radius;
    }

    @Override
    double calculateVolume() {
        return (4.0 / 3.0) * Math.PI * radius * radius * radius;
    }

    @Override
    double calculateSurfaceArea() {
        return 4 * Math.PI * radius * radius;
    }
}
```

\\

```
   class  Cube extends Shape3D {
      private double side;

      Cube(double side) {
         this.side = side;
      }

      @Override
      double calculateVolume() {
         return side * side * side;
      }

      @Override
      double calculateSurfaceArea() {
         return 6 * side * side;
      }
   }

   class main {
      public static void main(String[] args) {
        System.out.println("NAME:j.bhagiradha ,ROLL NO
:AV.SC.U4CSE24132,SEC:CSE-B");
         Sphere s = new Sphere(6.4);
         System.out.println("Volume of Sphere: " + s.calculateVolume());
         System.out.println("Surface Area of Sphere: " +
s.calculateSurfaceArea());

         Cube c = new Cube(4.5);
         System.out.println("Volume of Cube: " + c.calculateVolume());
         System.out.println("Surface Area of Cube: " + c.calculateSurfaceArea());
      }
   }
```

Output :

\\

Error table :

S.NO	Error Name	Error Rectification
1	Main Class	Better to create main class name same as the file you saved and first letter is capital.
2	Data type	As per need provide data type
3	Abstract method	Implementation in subclass only

3) **Aim :** write a java program using an abstract class to define a method for pattern printing Create an abstract class named pattern printer with an abstract method print pattern (int n) and a concrete method to display the pattern tittle. Implement two sub class :
1.star pattern -prints a right-angled triangle of stars(*)
2.number pattern-prints a right angled triangle of increasing numbers.
In the main () method,create objects of both subclasses and print the pattern for a given number of rows.

Important points :

1.Here we used nested for loop concept the block of code is executed until the condition is false.

2.Here we used abstract class concept it is a restricted class that cannot be instantiated (cannot have objects created directly) and is typically designed to be extended by subclasses.

Program :

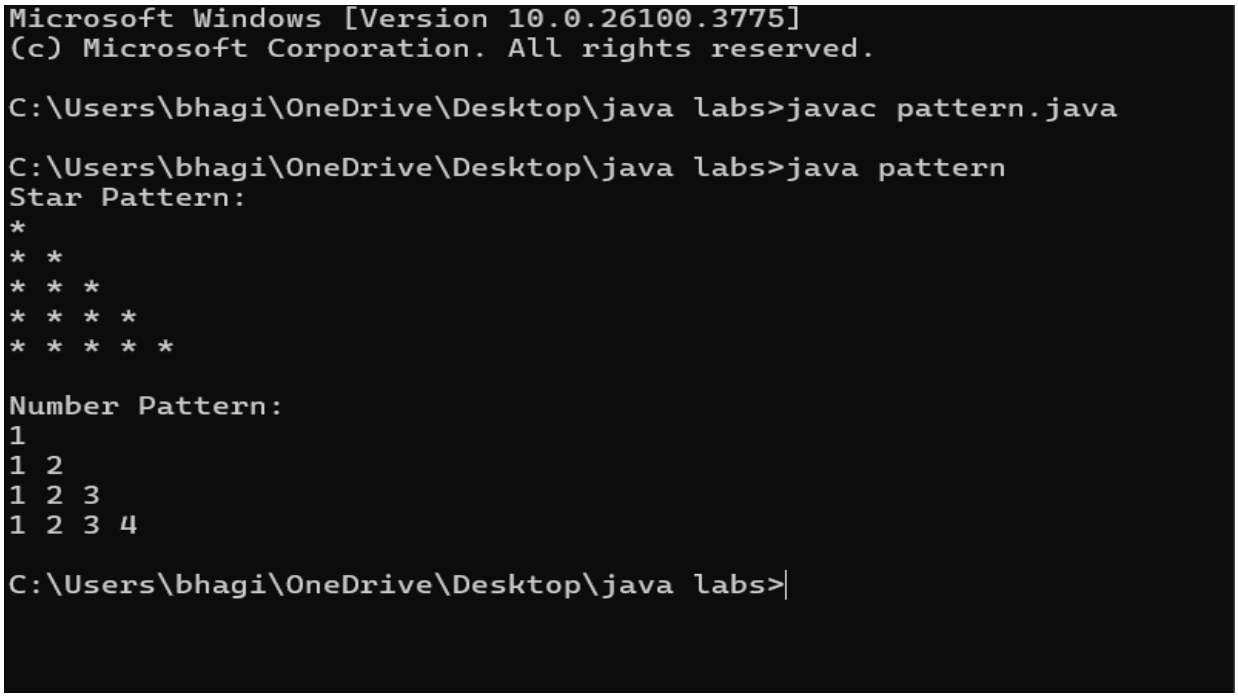
```
abstract class PatternPrinter{     public abstract void printPattern(int n);     public
void printTitle(String title) {
        System.out.println(title);
    }
}
class StarPattern extends PatternPrinter {
```

\\

```
    @Override
    public void printPattern(int n){

        for (int i=1;i<=n;i++) {            for (int j=1;j<=i;j++)
{                   System.out.print("* ");
            }
            System.out.println();
        }
    }
}
class NumberPattern extends PatternPrinter {
    @Override
    public void printPattern(int n) {

        for (int i=1;i<n;i++) {            for (int j=1;j<=i;j++)
{                   System.out.print(j+" ");
            }
            System.out.println();
        }



    }
}

public class pattern {
    public static void main(String[] args) {        int rows=5;
        PatternPrinter starPattern = new StarPattern();
        PatternPrinter numberPattern = new NumberPattern();
        starPattern.printTitle("Star
Pattern:");        starPattern.printPattern(rows);          System.out.println();
        numberPattern.printTitle("Number
Pattern:");        numberPattern.printPattern(rows);
    }
}
```

\\

Output :



Error table :

S.NO	Error Name	Error Rectification
1	Main Class	Better to create main class name same as the file you saved and first letter is capital.
2	Data type	As per need provide data type
3	Syntax in for	Initializing value and condition should be correct
4	overridding	Same method names