23CSE111

OBJECT ORIENTED PROGRAMMING

DOCUMENT

Department of computer science Engineering

Amrita School of Engineering

Amrita Vishwa Vidyapeetham, Amaravati Campus

Name: Davis J.Johney

Verified By :

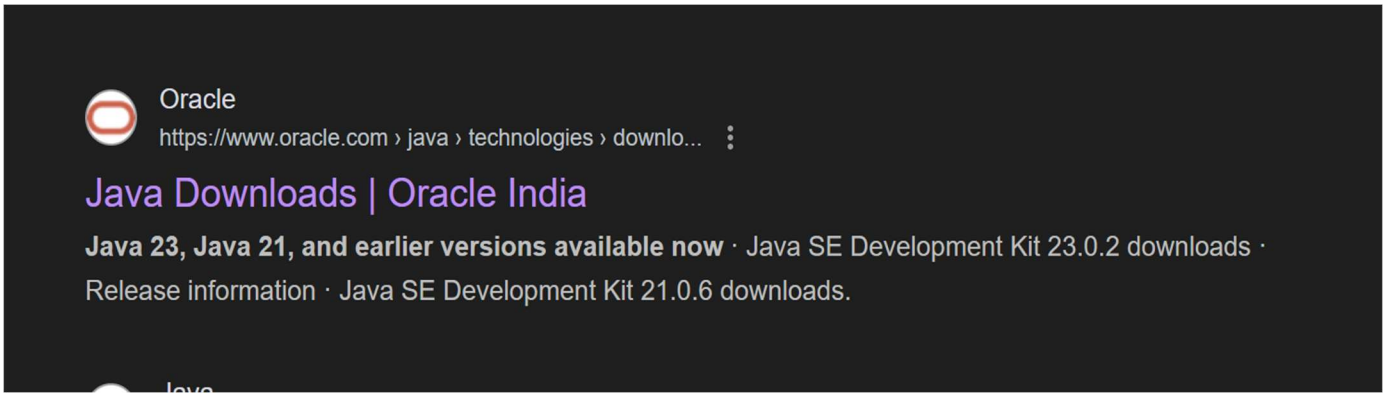
Roll No: CSEA-24041

WEEK - 1

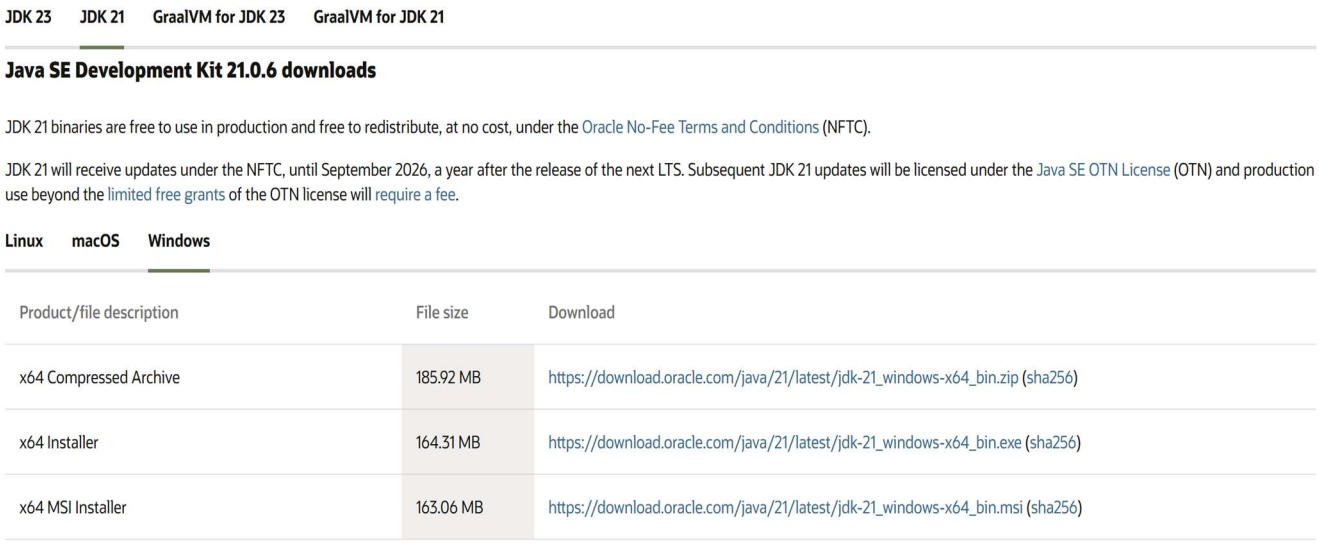
a) Write the steps to download and install Java.

Aim : Download and Install Java Software

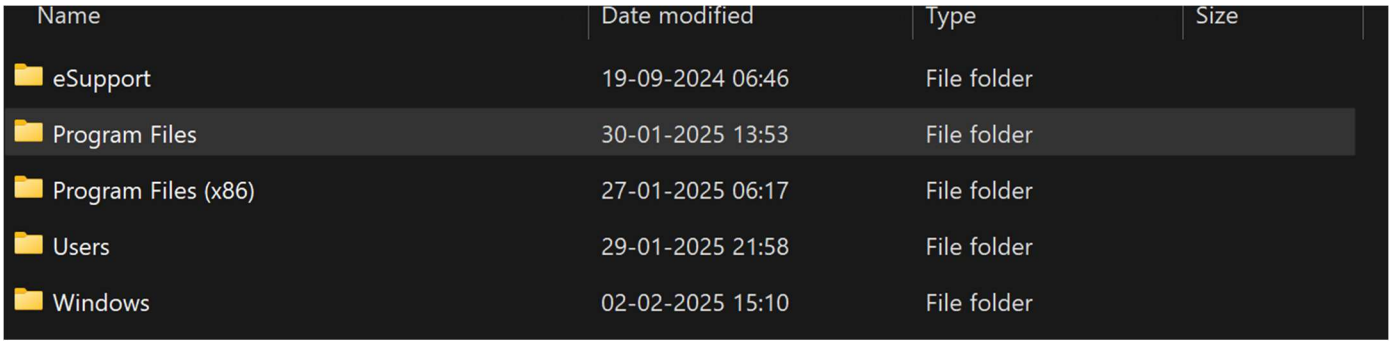
Step – 1 : Visit any web browser and search for java download. Select the official Oracle website.



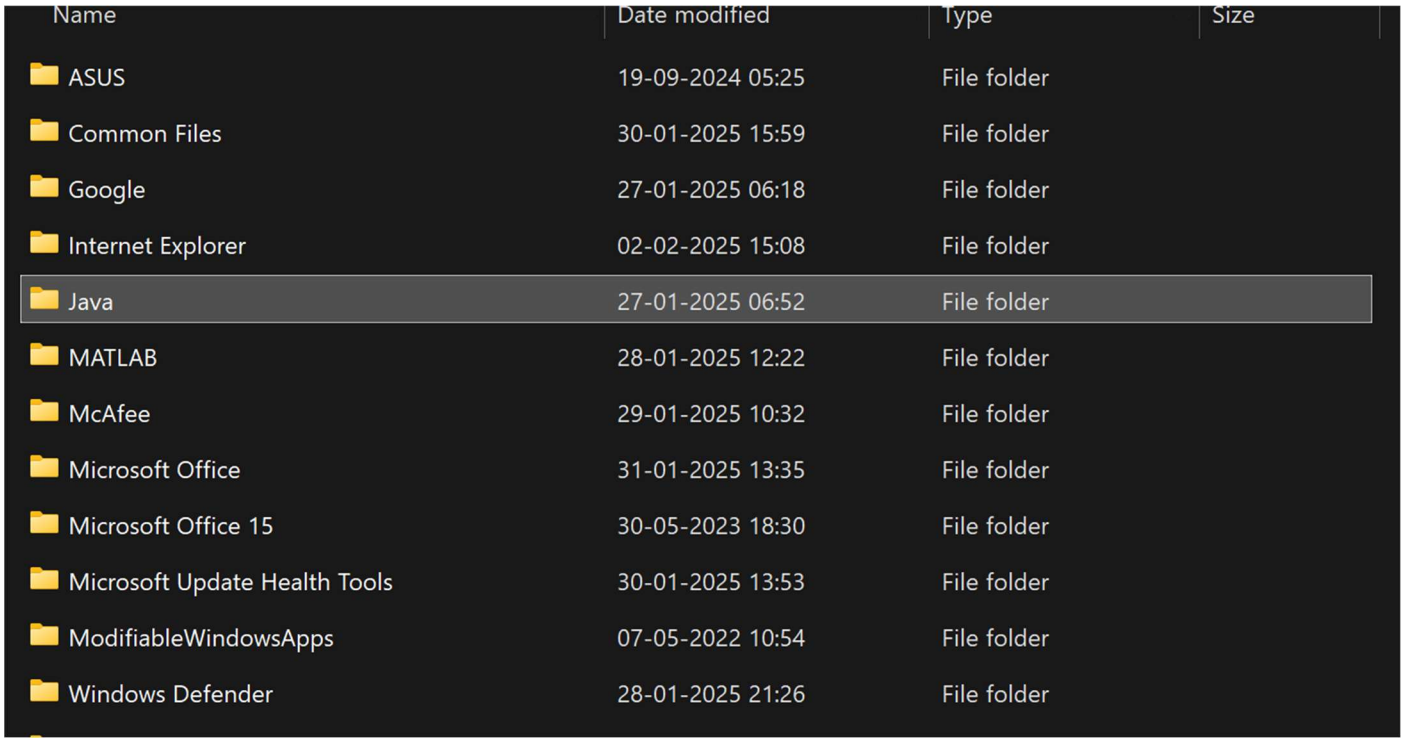
Step – 2 : Open Oracle website and select the LTS "JDK 21 " for Windows and select "X64 Installer" and download it.



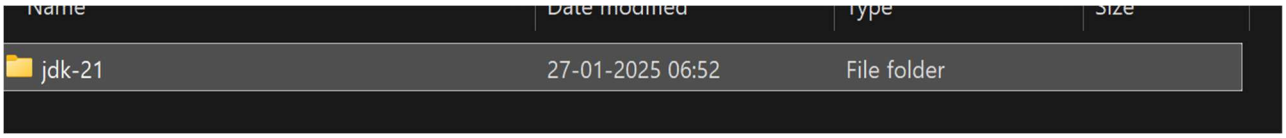
Step – 3 : After downloading open “C-drive” on your pc and select “Program Files”, open “JDK 21”



Name	Date modified	Type	Size
eSupport	19-09-2024 06:46	File folder	
Program Files	30-01-2025 13:53	File folder	
Program Files (x86)	27-01-2025 06:17	File folder	
Users	29-01-2025 21:58	File folder	
Windows	02-02-2025 15:10	File folder	

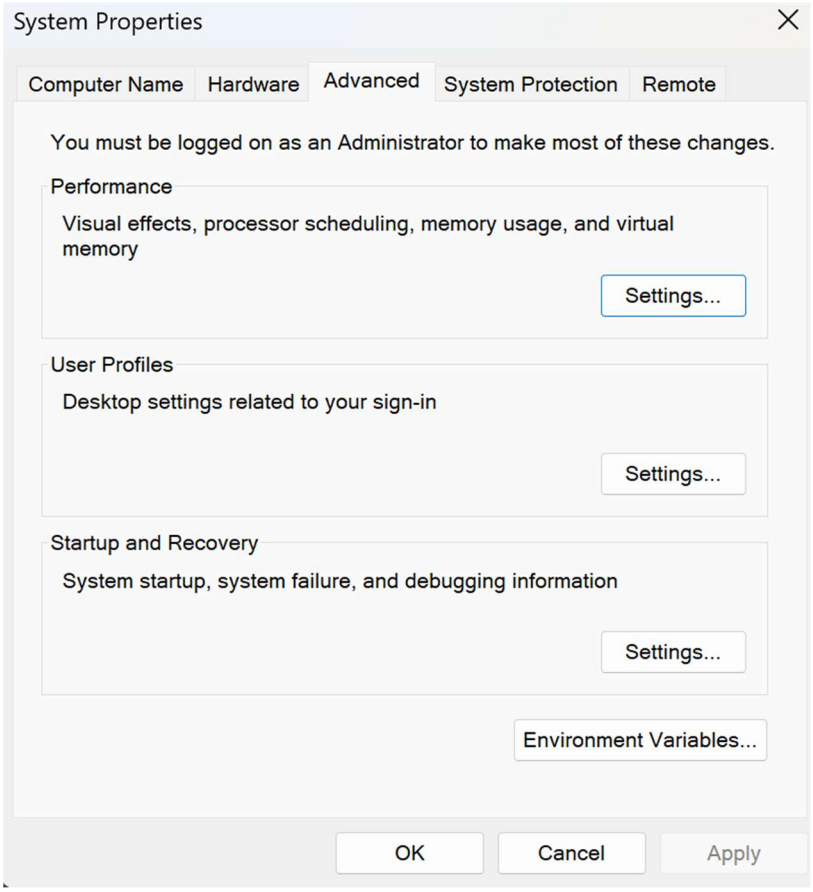


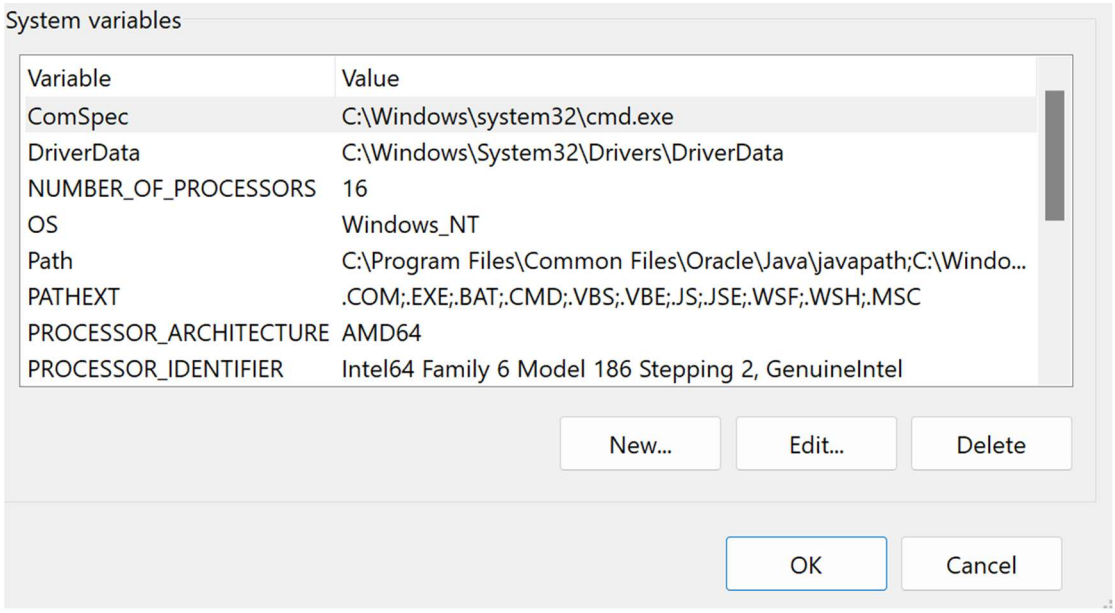
Name	Date modified	Type	Size
ASUS	19-09-2024 05:25	File folder	
Common Files	30-01-2025 15:59	File folder	
Google	27-01-2025 06:18	File folder	
Internet Explorer	02-02-2025 15:08	File folder	
Java	27-01-2025 06:52	File folder	
MATLAB	28-01-2025 12:22	File folder	
McAfee	29-01-2025 10:32	File folder	
Microsoft Office	31-01-2025 13:35	File folder	
Microsoft Office 15	30-05-2023 18:30	File folder	
Microsoft Update Health Tools	30-01-2025 13:53	File folder	
ModifiableWindowsApps	07-05-2022 10:54	File folder	
Windows Defender	28-01-2025 21:26	File folder	



Name	Date modified	Type	Size
jdk-21	27-01-2025 06:52	File folder	

Step – 4 : Open environmental variables and add a new file with path.





Step – 5 : Verify java version in command window

```
C:\Users\bhuva>java --version
java 21.0.6 2025-01-21 LTS
Java(TM) SE Runtime Environment (build 21.0.6+8-LTS-188)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.6+8-LTS-188, mixed mode, sharing)

C:\Users\bhuva>
```

b) Write a java program to print the message "Welcome to java programming".

Code:

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

    }
}
```

```
C:\Users\bhuva\OneDrive\desktop\JAVA>javac ex1.java

C:\Users\bhuva\OneDrive\desktop\JAVA>java ex1
Welcome to java programming.

C:\Users\bhuva\OneDrive\desktop\JAVA>
```

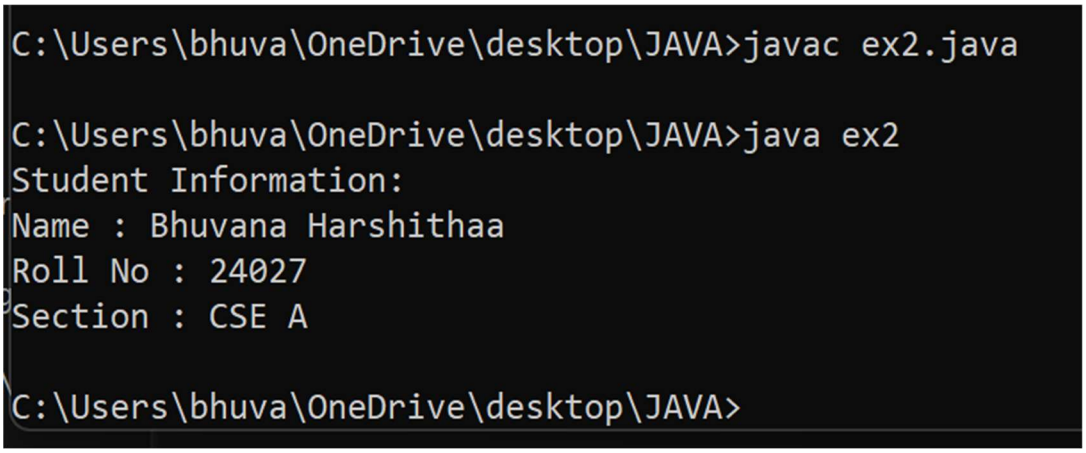
Error :

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

c) Write a java program to print the student information

Code :

```
class ex2{
        public static void main(String[] args){
            System.out.println("Student Information:");
            System.out.println("Name : Bhuvana Harshithaa");
            System.out.println("Roll No : 24027")
            System.out.println("Section : CSE  A")

    }

}
```



ERRORS :

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

WEEK – 2

a) Write a java program to clalculate area of rectangle.

Code :

```
import java.util.Scanner;
    public class arear{
      public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a value : ");
        int b = input.nextInt();
        System.out.print("Enter a value : ");
        int l = input.nextInt();
          int area = b*l;
         System.out.print("The area of the rectangle is : "+ area);
   input.close();
  }
}
```

Output :

```
C:\Users\bhuva\OneDrive\desktop\JAVA>java arear
Enter a value : 10
Enter a value : 25
The area of the rectangle is : 250
C:\Users\bhuva\OneDrive\desktop\JAVA>
```

ERRORS :

S.No	Expected Error	Reason
1	;	; is expected at end
2	area	Declaration of int type variable

b) Write a java program to convert temperature from Celsius to Fahrenheit and vice versa.

Code :

```
import java.util.Scanner;


class temp{
 public static void main(String[] args){
  Scanner input =new Scanner(System.in);
  System.out.print("enter the the temperature in degrees:");
   double deg=input.nextDouble();
 System.out.println("the temperatuer in fahrenheit"+((deg*9/5)+32));

  }

}
```

Output :

```
C:\Users\bhuva\OneDrive\desktop\JAVA>javac temp.java

C:\Users\bhuva\OneDrive\desktop\JAVA>java temp
enter the the temperature in degrees:100
the temperatuer in fahrenheit212.0

C:\Users\bhuva\OneDrive\desktop\JAVA>
```

ERRORS :

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

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

Code :

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

Output :

```
C:\Users\bhuva\OneDrive\desktop\JAVA>java si
Enter principal amount : 10000
Enter rate of interest : 2
Enter the time period : 3
The simple Interest is : 600
C:\Users\bhuva\OneDrive\desktop\JAVA>
```

ERRORS :

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

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

Code :

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

 }
```

Output :

```
C:\Users\bhuva\OneDrive\desktop\JAVA>javac largest.java

C:\Users\bhuva\OneDrive\desktop\JAVA>java largest
Enter number a : 10
Enter number b : 15
Enter number c : 6
The largest number is : 15
C:\Users\bhuva\OneDrive\desktop\JAVA>
```

ERRORS :

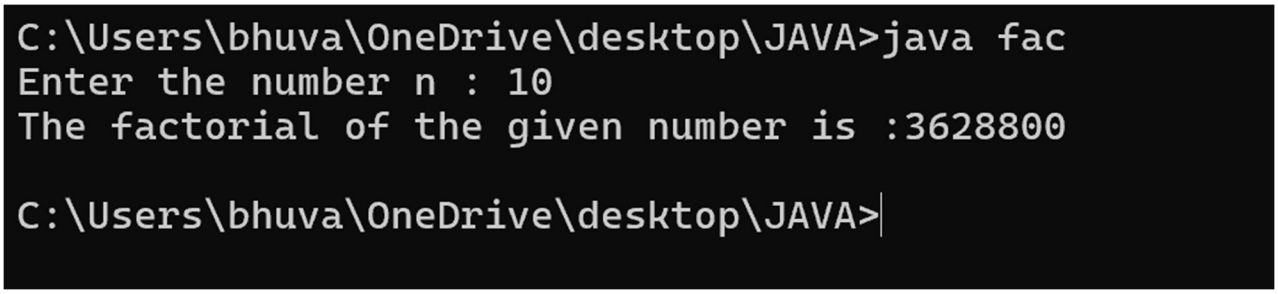
S.No	Expected Error	Reason
1	?	Checks the condition
2	:	Comparing between two variables

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

Code :

```
import java.util.Scanner;
public class fac{
  public static void main(String[] args){
  Scanner input = new Scanner(System.in);
  System.out.print("Enter the number n : ");
  int n = input.nextInt();
  int fac = 1;
 for(int i = 2; i<=n;i++){
 fac *= i;
}
      System.out.println( "The factorial of the given number is :" + fac);
      input.close();

  }
}
```

Output :



ERRORS :

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

WEEK – 3

a) Create the java program with the following instructions

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

Code:

```
public class Car{
public String carColor;
private String carBrand;
private String fuelType;
public int mileage;
Car(String carColor , String carBrand , String fuelType , int mileage){
this.carColor =  carColor;
this.carBrand = carBrand;
this.fuelType = fuelType;
this.mileage = mileage;
System.out.println(carColor + " " + carBrand + " " + fuelType + " " + mileage);
}
public void Start(){
System.out.println("The car has just started");
}
public void Stop(){
System.out.println("The car has just stopped");
}
public void Service(){
System.out.println("The car is in good condition");
 }
public static void main(String[] args){
Car Car1 = new Car("Black","Hyundai","Petrol",100);
Car Car2 = new Car("White","Suzuki","Diesel",150);
Car Car3 = new Car("Red","Benz","Petrol",200);
Car1.Start();
}
}
```

Output :

```
C:\Users\bhuva\OneDrive\desktop\JAVA>javac Car.java

C:\Users\bhuva\OneDrive\desktop\JAVA>java Car
Welcome to Car garage
Welcome to Car garage
Welcome to Car garage
The car has just started
```

Errors :

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

Class Diagram :

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

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

i) In deposit( ) whenever an amount is deposited it has to be updated with current amount
ii) In withdraw( ) whenever an amount is withdrawn it has to be less than current amount else print "Insufficient funds".

Code :

```
public class BankAccount{
private String Name;
private int AccNo, CurrBal ;
BankAccount(String Name, int AccNo, int CurrBal){
this.Name = Name;
this.AccNo = AccNo;
this.CurrBal = CurrBal;
System.out.println("The customers are : " + this.Name + " ");
}
public int deposit(int dAmt){
CurrBal = CurrBal + dAmt ;
return CurrBal;
}
public void withdraw(int wAmount){
if(wAmount < CurrBal){
CurrBal = CurrBal - wAmount ;
System.out.println(CurrBal);
}
else{
System.out.println("Insufficient funds");
}
}
public static void main(String[] args){
BankAccount Bhuvana = new BankAccount("Bhuvana",1500,10000);
Bhuvana.withdraw(13000);
Bhuvana.withdraw(1900);
int FinalAmount = Bhuvana.deposit(10000);
System.out.println(FinalAmount);
}
```

Output :

```
C:\Users\bhuva\OneDrive\desktop\JAVA>javac BankAccount.java

C:\Users\bhuva\OneDrive\desktop\JAVA>java BankAccount
The customers are : Bhuvana
Insufficient funds
8100
18100
```

Errors :

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

Class Diagram :

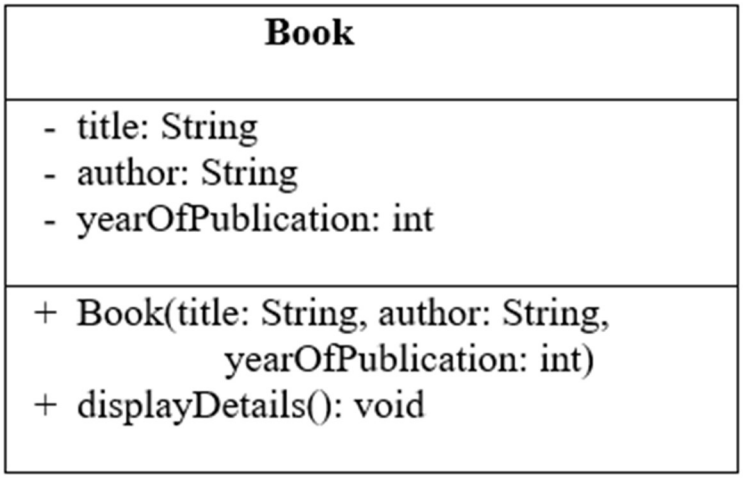
BankAccount
- Name : String - AccNo : String - CurrBal : String
+ BankAccount( ) : void + deposit( ) : int + withdraw( ) : void

WEEK – 4

a) Write a java program with class named "Book". The class should contain various attributes such as "Title of the book , author , year of publication ". It should also contain a constructor with parameters which initializes " Title of the book, author, year of publication". Create a method which displays the details of the book. i.e. " Title of the book, author and year of publication". Display the details of two books by creating two objects.

Code :

```
class Book{
// beginning of the class book
public String Title;
private String author;
public int yearOfPublication;
// beginning of constructor
Book(String Title , String author , int yearOfPublication){
this.Title = Title;
this.author = author;
this.yearOfPublication = yearOfPublication;
}
//constructor ends here
// methos display starts here
 public void display(){
System.out.println("Title of the book is : " + Title + "The name of the author is : " + author +
 "The  year of publication is : " + yearOfPublication );
}
// method display ends here
// creating objects
public static void main(String[] args){
Book Book1 = new Book("Harry Potter" , "J.K.Rowling" ,1993);
Book Book2 = new Book("Someone Like You" , "Nikitha Singh" , 2010);
Book1.display();
Book2.display();
}
}
// class ends here
```



Output :

```
C:\Users\bhuva\OneDrive\desktop\JAVA>javac Book.java

C:\Users\bhuva\OneDrive\desktop\JAVA>java Book
Title of the book is : Harry PotterThe name of the author is : J.K.RowlingThe year of publication is : 1993
Title of the book is : Someone Like YouThe name of the author is : Nikitha SinghThe year of publication is : 2010
```

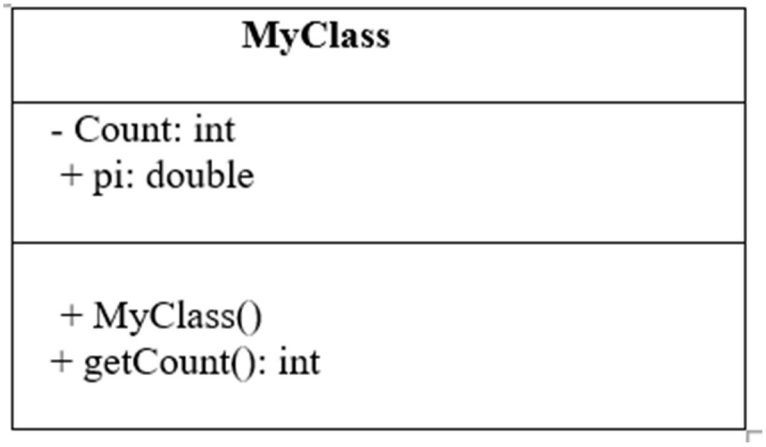
Errors :

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

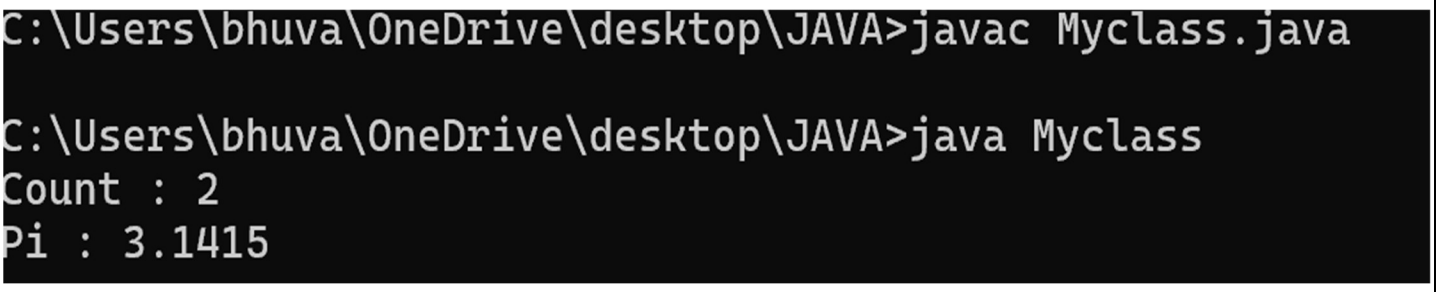
b) To create a java program with class named Myclass with a static variable "Count" of "int type", Initialized to 0 and a constant variable "pi" of type double , initialized to 3.1415 as attributes of that class Now, define a constructor for "Myclass" that increments the "Count" variable each that an object of Myclass is created. Finally , print the final values of "Count" and "pi" variables .

Code :

```
class Myclass{
// class starts here
static int Count = 0;
final double pi = 3.1415;
// the constructor starts here
Myclass(){
Count++;
}
// the constructor ends here
public static void main(String[] args){
Myclass c1 = new Myclass();
Myclass c2 = new Myclass();
System.out.println("Count : " + c1.Count);
System.out.println("Pi : " + c1.pi);
}
}
// class ends here
```



Output :



Errors :

S.No.	Expected Error	Reason
1	.variable	We must mention variable name to call the variable
2	static	Static variables contain only one value

WEEK 5

a) Create a calculator using the operations including addition using subtraction multiplication and division using multilateral inheritance and display the desired output.

Code-

```
class calculator {
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 calculator {
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 main {
```

```
public static void main(String[] args) {
aacalc c = new aacalc(10, 2);
c.divi();
c.mult();
c.add();
}
}
```

Basic Operations
+ add (a,b) +subtract (a,b)

Multiplication
+Multiply (a,b)

Division
+ Divide (a,b)

Subtraction
+ subtraction(a,b)

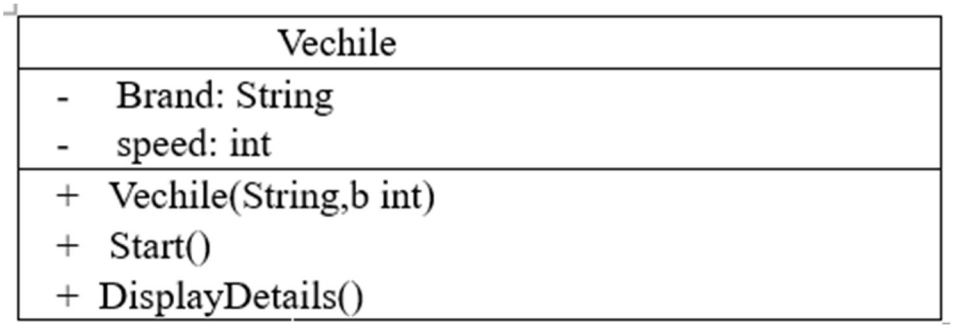
Calculator
+calculate (op,a,b)

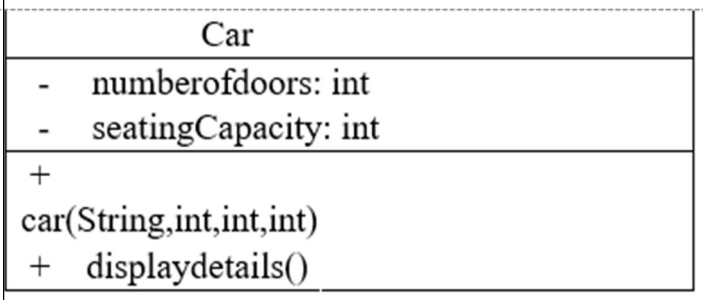
S.NO	Errors	Rectification
1	.variable	We must mention variable name to call the variable.
2	static	Static variables contain only one value.

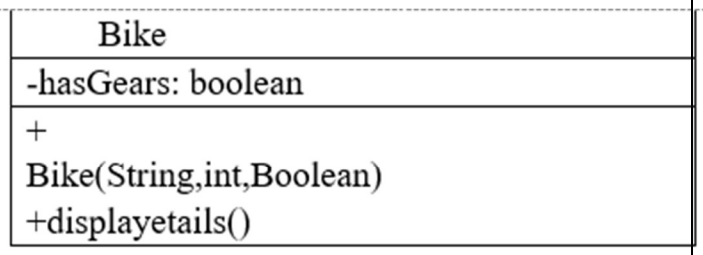
```
C:\Users\johne\OneDrive\Desktop\Java assignment>java main
Division: 5.0
Multiplication: 20
Difference: 8
Sum: 12

C:\Users\johne\OneDrive\Desktop\Java assignment>
```

b) A Vechile 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 a need a program to store details about each vehicle such as brand and speed .

Cars should have an additional properties .
"Number of doors " seating capacity.
Bikes should have a property indicating whether they have gears are not ?
The system should also include a fuction to display details about each vehicle and indicate when a vechicle is starting .
If the company describes to add a new type of vechile 'truck' how would you modify above program.
Truck should include an addition property capacity 'in tons'.
Create a show truck details method to display the trucks capacity.
Write a constructor for truck that initializes all properties.
Implement the truck class and update the main method to create a truck object and also create an object and also create an object car and bike subclass find display it details.







Truck
-capacity: double
+ truck(String,int,double) +showtruckdetails() +displaydetails()

Code-

```
                // Base class for Vehicle
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");
    }
}

// Car class that extends Vehicle
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);
    }
}
```

```
// Bike class that extends Vehicle
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"));
    }
}

// Truck class that extends Vehicle
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();
    }
}

// Main class to test the implementation
public 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();
    }
}
```

```
E:\JAVA>java Main2
Toyota is starting.
Brand: Toyota
Speed: 180 km/h
Number of Doors: 4
Seating Capacity: 5

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

Volvo is starting.
Brand: Volvo
Speed: 100 km/h
Truck Capacity: 10.5 tons
```

S.NO	Errors	Rectification
1	.variable	We must mention variable name to call the variable.
2	static	Static variables contain only one value.