

23CSE111

OOPS

LAB MANUAL



Department of CSE

Amrita School of Engineering

Amrita Vishwa Vidyapeetham, Amaravati Campus

Verified By

**Name: E. Hima Teja Goud
Roll No: 24048**

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

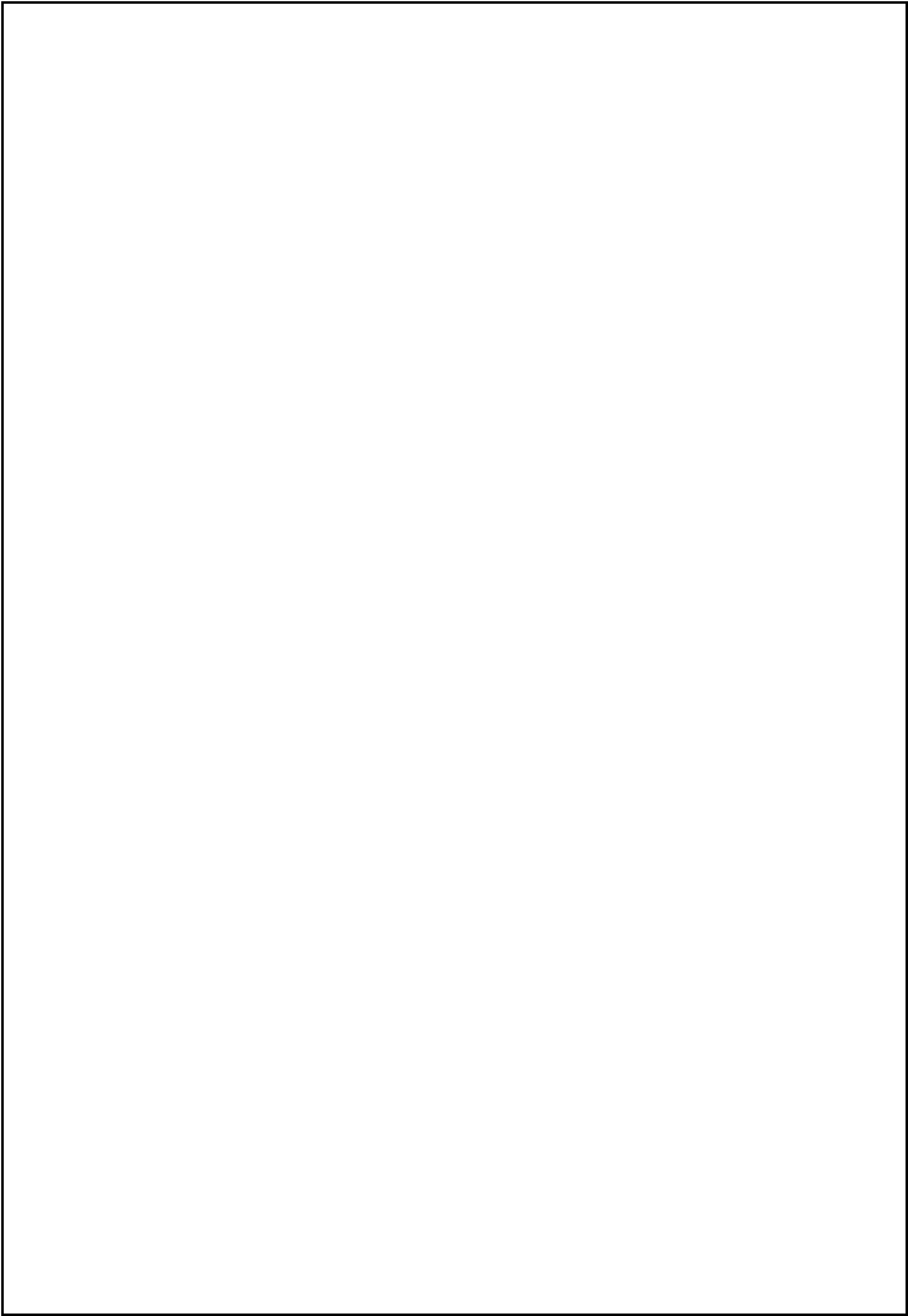
	Creating a BankAccount class with the given instructions			
WEEK 4		02-03-2025		
1	Write a java program with class named "Book" with given instructions.			
2	To create a java program with class named Myclass with given instructions.			
WEEK 5		09-03-2025		
1	Create a calc using the operations including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

4	Create a shape class with method calculateArea() that is overloaded for different shapes (eg: square, rectangle).Then create a subclass Circle that overrides calculateArea() method for circle.			
WEEK 7		14-04-2025		
1	Write a java program to create an abstract class Animal with abstract method sound and create subclasses Lion and Tiger that implements the method.			
2	Write a java program to create an abstract class shape3D with abstract methods to calculate volume and surfacearea and create subclasses for sphere and cube that implements these methods.			
3	Create an abstract class PatternPrint with an abstract method printing to print the pattern and a concrete method to display the pattern . Implement the patterns 1. Star Pattern - prints a right angled triangle of stars 2. Number Pattern – prints a right Angled triangle of increasing numbers.			
WEEK 8		21-04-2025		
1	Write a java program creating an interface Shape with the get perimeter method create classes rectangle,triangle and circle that implements the shape interface ,implement the getperimeter method for each of the three classes			
2	Write a java program to create an interface playable with a method play() that takes no arguments and returns void create three classes football,volleyball and			

	basketball that implements the playable			
	and override the play method to play the respective sports.			
3	Write a java program to implement a login system using interfaces			
WEEK 9		28-04-2025		
1	Write a java program to create a method that takes integer as a parameter and throws an exception if the number is even.			
2	Write a java program to create a method that reads a file and throws an exception if the file is not found.			
3	Write a java program to handle arithmetic exception using try catch and finally.			
4	<p>Java program to stimulate a university system using inner classes</p> <p>.Create an outer class named University with a variable Universityname</p> <p>.Inside it define two non static classes</p> <p>1. Department - with variable like deptName and deptCode and a method to display department details.</p> <p>2. Student – variable like stdName and stdCode and a method to display Student Details</p>			

WEEK 10		28-04-2025		
1	Write a java program to generate a password for a student using his/her initials and age. The password displayed should be the string			
	consists of first character of first name, middle name, last name with age.			
2	Design and implement a java program that will do the following operations to the string "welcome! You are practicing strings concept". -convert all alphabets to capital letters and print the result -convert all alphabets to lower-case letters and print out the result -print out the length of string -print out the index of concept			
3	Implement a java program using below array methods. -sorting the elements(numbers and strings) of an array -convert the array elements into string -fill the part of an array -copy the elements of one array into another			
4	Implement a java program using the below array list methods -insert an element at a particular index in the array list -modify an element in the array list			

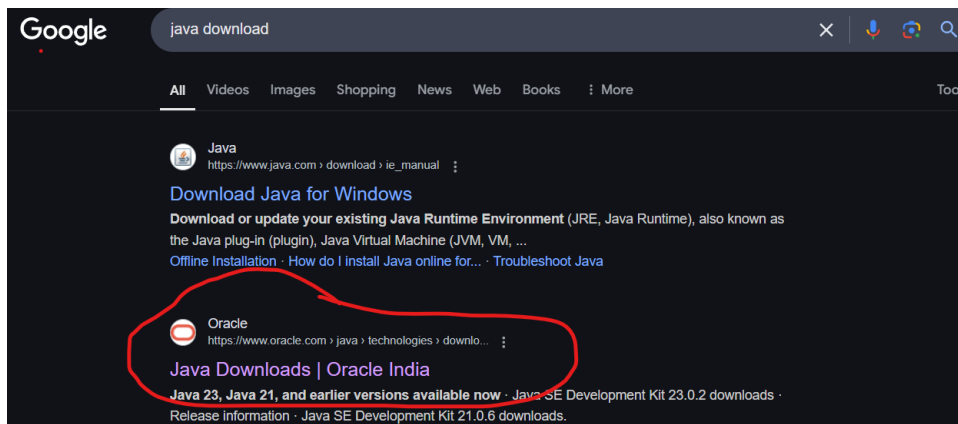
	<ul style="list-style-type: none">-access an element from the array list-remove an element from array list-clear the elements from the array list			
--	---	--	--	--



WEEK-1

➤ Program : 1

- Aim : Download and Install Java Software.
- Step 1 : Visit chrome and search “java download”.And select Oracle website.



- Step 2 : Now open Oracle website scroll down and now select “JDK 21” for Windows and select “X64 installer” and download it.

JDK 23	JDK 21	GraalVM for JDK 23	GraalVM for JDK 21
Java SE Development Kit 23.0.2 downloads			
JDK 23 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC) .			
JDK 23 will receive updates under these terms, until March 2025, when it will be superseded by JDK 24.			
Linux	macOS	Windows	
Product/file description	File size	Download	
x64 Compressed Archive	228.77 MB	https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.zip (sha256)	
x64 Installer	205.27 MB	https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.exe (sha256)	
x64 MSI Installer	204.02 MB	https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.msi (sha256)	

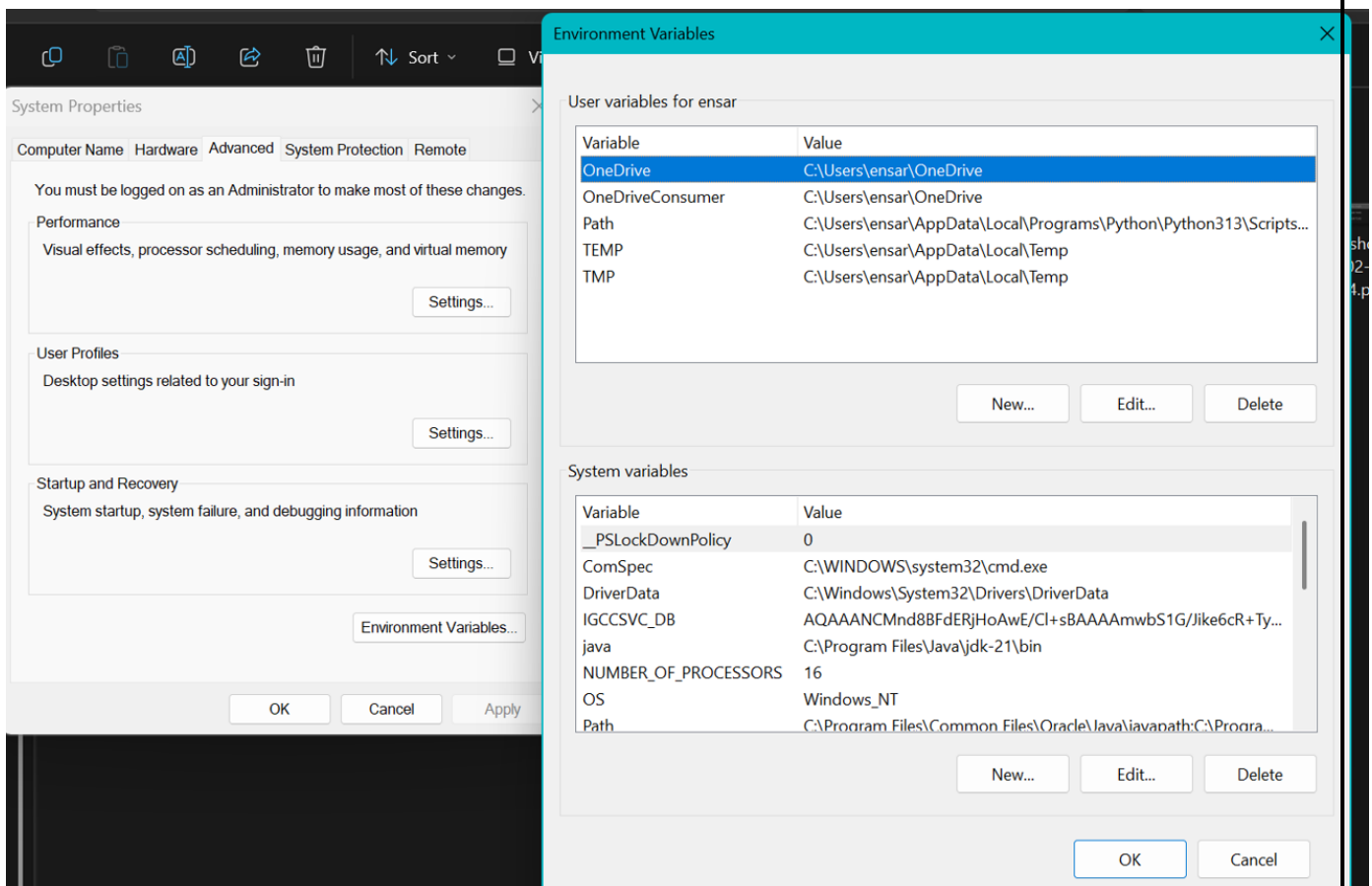
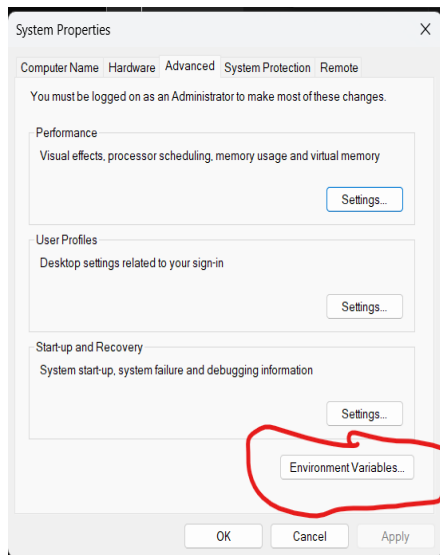
- Step 3 : After downloading open “this pc” in our laptop and open “program files”,open “java”,open “JDK 21”

LDPlayer	20-07-2024 14:04	File folder
OneDriveTemp	07-11-2024 21:06	File folder
PerfLogs	01-04-2024 12:56	File folder
Program Files	24-12-2024 20:28	File folder
Program Files (x86)	21-01-2025 18:08	File folder
ProgramData	26-12-2024 21:06	File folder

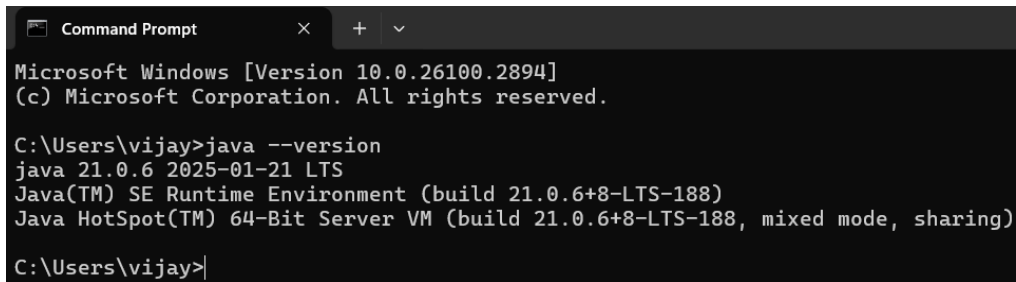
Internet Explorer	24-12-2024 10:53	File folder
Java	27-01-2025 15:21	File folder
MATLAB	30-09-2024 19:47	File folder
McAfee	20-06-2024 09:24	File folder
jdk-21	27-01-2025 15:21	File folder
jre1.8.0_421	23-09-2024 19:43	File folder
latest	23-09-2024 19:43	File folder

This PC > OS (C:) > Program Files > Java > jdk-21 >			
<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>Sort</div> <div>View</div> <div></div> </div>			
Name	Date modified	Type	Size
bin	27-01-2025 15:21	File folder	
conf	27-01-2025 15:21	File folder	
include	27-01-2025 15:21	File folder	
jmods	27-01-2025 15:21	File folder	
legal	27-01-2025 15:21	File folder	
lib	27-01-2025 15:21	File folder	
LICENSE	27-01-2025 15:21	File	7 KB
README	27-01-2025 15:21	File	1 KB
release	27-01-2025 15:21	File	2 KB

- Step 4 : In the task bar search and open “environment variables of system”,after opening environment variables, go to the system variables and see for java if there leave it. Or click path and add “JAVA” in ‘variable name’ and copy link in ‘variable value’



- Step 5 : Verifying Installation of Java. Again open task bar and search “cmd”, open it and type “java –version” and press enter. It will show the version of installation of java.



```
Command Prompt
Microsoft Windows [Version 10.0.26100.2894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vijay>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\vijay>
```

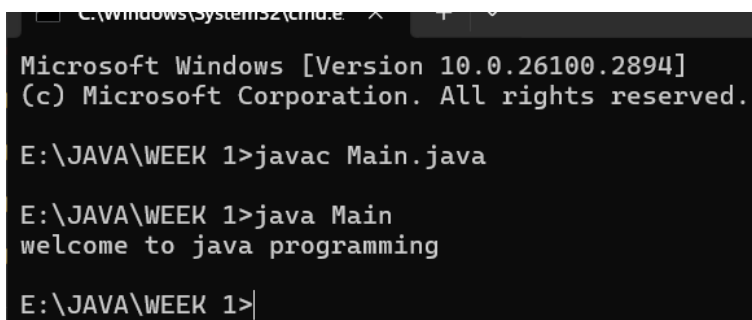
Successfully Java is installed and it will show the version otherwise it will show error and command is not recognized.

➤ Program : 2

Q) Write a java program to print the message “welcome to java program”.

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

OUTPUT



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.26100.2894]
(c) Microsoft Corporation. All rights reserved.

E:\JAVA\WEEK 1>javac Main.java

E:\JAVA\WEEK 1>java Main
welcome to java programming

E:\JAVA\WEEK 1>
```

➤ Program : 3

Q) Write a java program that prints name,roll number,section of a student.

```
public class my_profile{
    public static void main(String[] arg){
        System.out.println("name:E.Hima Teja Goud");
        System.out.println("sec:CSE-A");
        System.out.println("Roll.no:24048");
    }
}
```

OUTPUT

```
Microsoft Windows [Version 10.0.26100.2605]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ensar\OneDrive\Desktop\JAVA\week1>javac my_profile.java

C:\Users\ensar\OneDrive\Desktop\JAVA\week1>java my_profile
name:E.Hima Teja Goud
sec:CSE-A
Roll.no:24048

C:\Users\ensar\OneDrive\Desktop\JAVA\week1>
```

WEEK-2

➤ Program : 1

```
import java.util.Scanner;

class rectangle{
    public static void main(String[]args){
        Scanner input=new Scanner(System.in);
        System.out.println("enter the length");
        int len=input.nextInt();
        System.out.println("enter the bredth");
        int bred=input.nextInt();
```

```
int area=len*bred;
System.out.println(area);
}}
```

```
enter the length
20
enter the bredth
10
200
```

OUTPUT:

S.no	EXPECTED ERROR	REASON
1.	;	; is expected at end
2.	AREA	Declaration of int type variable

➤ Program : 2

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

```
enter the the temperature in degrees:
45
```

OUTPUT: the temperatuer in fahrenheit113.0

ERRORS:

S.No	EXPECTED ERROR	REASON
1.	;	;expected at end
2.	Input().close	The input is expected to closed

➤ Program : 3

```
import java.util.Scanner;
class simpleintrest{
    public static void main(String[]args){
        Scanner input=new Scanner(System.in);
        System.out.println("enter the p value");
        int p=input.nextInt();
        System.out.println("enter the t value");
        int t=input.nextInt();
        System.out.println("enter the r value");
```

```

int r=input.nextInt();
float si=(p*t*r)/100;
System.out.println(si);
}
}

```

```

enter the p value
10000
enter the t value
10
enter the r value
34
34000.0

```

OUTPUT:

S.No	EXPECTED ERROR	REASON
1.	;	; is expected at end
2.	Int t	Without declaring t the compiler cannot execute the program.

➤ Program : 4

```

import java.util.Scanner;
class largest{
public static void main(String[]args){
Scanner input=new Scanner(System.in);
System.out.println("enter value of A");
int a=input.nextInt();
System.out.println("enter value of B");
int b=input.nextInt();
System.out.println("enter value of C");
int c=input.nextInt();
int large=(a>b)?((a>c)?a:c):((b>c)?b:c);
System.out.println(large);
}
}

```

OUTPUT:

```

enter value of A
23
enter value of B
13
enter value of C
24
24

```

ERRORS:

S. No	EXPECT ED ERROR	REASO N
----------	-----------------------	------------

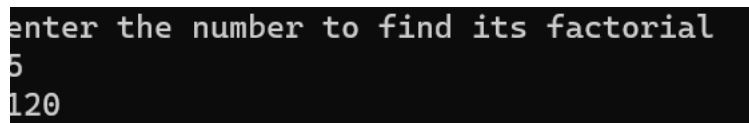
1.	?	Checks the condition
2.	:	Comparing between two variables

➤ **Program : 5**

```
import java.util.Scanner;
class factorial{
    public static void main(String[]args){
        Scanner input=new Scanner(System.in);
        System.out.println("enter the number to find its factorial");
        int n=input.nextInt();
        int sum=1;
        for(int i=1;i<=n;i++){
            sum=sum*i;}
        System.out.println(sum);
    }
}
```

OUTPUT:

ERRORS:



S.No	EXPECTED ERRORS	REASON
1.	}	To close for loop
2.	;	; expected

WEEK-3

- Program : 1

Q) Write a java program with the following instructions.

- Create a class with name car.
- Create four attributes named car_colour,car_brand,fuel_type,top_speed.
- Create three method named "Start_Racing","End_Race".{ }
- Create three objects named Car1,Car2,Car3.
- Create a constructor which should print "Welcome to Garage".

Class Diagram:

Car
- carColor: String
- carBrand: String
- fuelType: String
- topSpeed: int
+ Car(String,String,String,int)
+ startRacing()
+ endRace()

```
// Car.java
public class Car {
    // Attributes
    private String carColour;
    private String carBrand;
    private String fuelType;
    private int topSpeed;

    // Constructor
    public Car(String carColour, String carBrand, String fuelType, int topSpeed) {
        this.carColour = carColour;
        this.carBrand = carBrand;
        this.fuelType = fuelType;
        this.topSpeed = topSpeed;
        System.out.println("Welcome to car garage");
    }

    // Method to start racing
    public void startRacing() {
```

```

        System.out.println(carBrand + " (" + carColour + ") is starting the race with a top
speed of " + topSpeed + " km/h and runs on " + fuelType + "!");
    }

```

```

// Method to end race

```

```

public void endRace() {
    System.out.println(carBrand + " (" + carColour + ") has finished the race!");
}

```

```

// Main method to create objects and demonstrate functionality

```

```

public static void main(String[] args) {
    // Creating three objects
    Car car1 = new Car("Red", "Ferrari", "Petrol", 200);
    Car car2 = new Car("Blue", "Tesla", "Electric", 250);
    Car car3 = new Car("Black", "BMW", "Diesel", 220);

```

```

// Starting and ending races

```

```

car1.startRacing();
car1.endRace();

```

```

car2.startRacing();
car2.endRace();

```

```

car3.startRacing();
car3.endRace();
}

```

```

}

```

```

C:\Users\ensar\OneDrive\Desktop\JAVA\week1>java Car
Welcome to car garage
Welcome to car garage
Welcome to car garage
Ferrari (Red) is starting the race with a top speed of 200 km/h and runs on Petrol!
Ferrari (Red) has finished the race!
Tesla (Blue) is starting the race with a top speed of 250 km/h and runs on Electric!
Tesla (Blue) has finished the race!
BMW (Black) is starting the race with a top speed of 220 km/h and runs on Diesel!
BMW (Black) has finished the race!

C:\Users\ensar\OneDrive\Desktop\JAVA\week1>

```

s. n o	ERRORS	RECTIFICAT ION
1 .	}	Close the loop

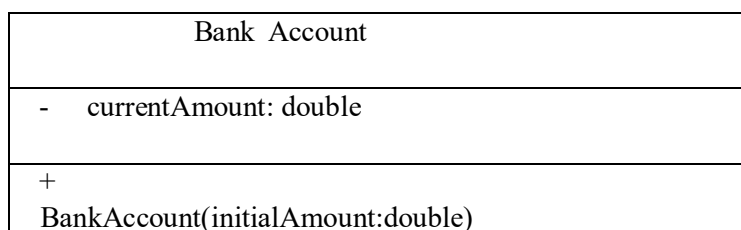
2 .	System.out.println	If we place the print statement inside the for loop it will print the each i value everytime but to print only the final value we must place it outside the for loop.
--------	--------------------	---

- Program : 2

Q) Write a class by writing java program named Bank Account with two methods “deposits and withdraw”.

- In deposit method whenever an amount is deposited it has to be updated with current amount (logic $C.A + D.A$).
- With draw amount whenever an amount is being withdraw it has to be less than the current amount less than the amount else print “Insufficient funds”.

Class Diagram:



+ deposit(amount: double):void
+ withdraw(amount: double):void
+ getCurrentAmount():double

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

```
class BankAccount {
    String name;
    int accountNumber;
    int currentBalance;

    // Constructor to initialize the bank account
    BankAccount(String name, int accountNumber, int currentBalance) {
        this.name = name;
        this.accountNumber = accountNumber;
        this.currentBalance = currentBalance;
        System.out.println("Customer Details: " + name + ", Account
Number: " + accountNumber + ", Current Balance: " + currentBalance);
    }

    // Method to withdraw an amount
    public void withdraw(int withdrawAmount) {
        if (withdrawAmount <= currentBalance) {
            currentBalance -= withdrawAmount;
            System.out.println("Withdrawn: " + withdrawAmount);
            System.out.println("Current Balance: " + currentBalance);
        } else {
            System.out.println("Insufficient Funds");
        }
    }
}
```

```
}

// Method to deposit an amount
public int deposit(int depositAmount) {
    currentBalance += depositAmount;
    System.out.println("Deposited: " + depositAmount);
    return currentBalance;
}

// Main method to run the program
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Input for account details
    System.out.print("Enter your name: ");
    String name = scanner.nextLine();
    System.out.print("Enter your account number: ");
    int accountNumber = scanner.nextInt();

    System.out.print("Enter your initial balance: ");
    int initialBalance = scanner.nextInt();

    // Create a new bank account
    BankAccount account = new BankAccount(name, accountNumber,
initialBalance);

    // Input for withdrawal and deposit
    System.out.print("Enter amount to withdraw: ");
    int withdrawAmount = scanner.nextInt();
    account.withdraw(withdrawAmount);

    System.out.print("Enter amount to deposit: ");
    int depositAmount = scanner.nextInt();
    account.deposit(depositAmount);

    // Final balance
    System.out.println("Final Amount: " + account.currentBalance);

    // Close the scanner
    scanner.close();
}
}
```

```

C:\Users\ensar\OneDrive\Desktop\JAVA\week1>java BankAccount
Enter your name: himateja
Enter your account number: 987654321
Enter your initial balance: 50000
Customer Details: himateja, Account Number: 987654321, Current Balance: 50000
Enter amount to withdraw: 200
Withdrawn: 200
Current Balance: 49800
Enter amount to deposit: 1000
Deposited: 1000
Final Amount: 50800

```

S.NO	Errors	Rectification
1	;	; is expected at end
2	Int t	Without declaring the compiler cannot execute the program.

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.

Class Diagram:

Book
<ul style="list-style-type: none">- title: String- author: String- yearOfPublication: int
<ul style="list-style-type: none">+ Book(title: String, author: String, yearOfPublication: int)

```
+ displayDetails():  
void
```

Code :

```
class Book {  
    // beginning of the class Book  
    public String title; // Changed Title to title for consistency  
    private String author;  
    public int yearOfPublication;  
    // beginning of constructor  
    Book(String title, String author, int yearOfPublication) {  
        this.title = title; // Changed Title to title for consistency  
        this.author = author;  
        this.yearOfPublication = yearOfPublication;  
    }  
    // constructor ends here  
    // method 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("solo leveling", "himateja", 2020);  
        Book book2 = new Book("naruto", "tej", 2010);  
        book1.display();  
        book2.display();  
    }  
}  
// class ends here
```

Output:

```
C:\Users\ensar\OneDrive\Desktop\JAVA\week1>java Book  
Title of the book is: solo leveling, The name of the author is: himateja, The year of publication is: 2020  
Title of the book is: naruto, The name of the author is: tej, The year of publication is: 2010
```

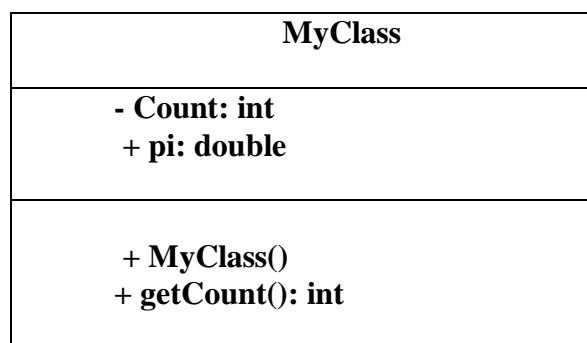
Errors :

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 .

Class Diagram:



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:

```
C:\Users\ensar\OneDrive\Desktop\JAVA\week1>java MyClass
Count : 2
Pi : 3.1415
```

Errors :

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

WEEK-5

- Program : 1

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

Class Diagram:

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)

CODE:

```
class bcalc {
    int a, b;
    int sum, diff;
    bcalc(int a, int b) {
        this.a = a;
        this.b = b;
    }
    public void add()
    { diff = a - b;
    sum = a + b;
    System.out.println("Difference: " + diff);
    System.out.println("Sum: " + sum);
    }
}
class acalc extends bcalc {
    int mul; acalc(int a, int b) {
        super(a, b);
    }
    public void mult() {
        mul = a * b;
        System.out.println("Multiplication: " + mul);
    }
}
class aacalc extends acalc {
    float div;
    aacalc(int a, int b) {
        super(a, b);
    }
    public void divi()
    {
        if (b != 0) { // Check to avoid division by zero
            div = (float) a / b;
```

```

System.out.println("Division: " + div);
}
else {
System.out.println("Division by zero error!");
}
}
}
}
class ocalc {

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

```

Output:

```

C:\Users\ensar\OneDrive\Desktop\JAVA\week1>java Main1
Addition (using subtraction): 8.0
Subtraction: 2.0
Multiplication: 15.0
Error! Division by zero.
Division: 2.5

```

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

- Program : 2

Q) 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.

Class Diagram:

Vechile
- Brand: String
- speed: int
+ Vechile(String,b int)
+ Start()
+ DisplayDetails()

Car
- numberofdoors: int
- seatingCapacity: int
+ car(String,int,int,int)
+ displaydetails()

Bike
-hasGears: boolean
+ Bike(String,int,Boolean)
+displayetails()

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

// 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();
}
}

```

Output:

```

C:\Users\ensar\OneDrive\Desktop\JAVA\week1>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.

WEEK-6

1) Write a java program to create a vehicles class with a method displayinfo() override this method in the car subclass to provide specific information about a car

- Company
- Model
- Price
- Seating capacity
- Petrol or not

Program:

```
public class vehicles {
    void displayinfo() {
        System.out.println("This is a vehicle");
    }
}

class Car extends vehicles {
    String Name, Model, Capacity;
    int Price;
    boolean Petrol;

    Car(String Name, String Model, String Capacity, int Price, boolean
    Petrol) {
        this.Name = Name;
        this.Model = Model;
        this.Capacity = Capacity;
        this.Price = Price;
        this.Petrol = Petrol;
    }

    @Override
    public void displayinfo() {
        System.out.println("Car name is: " + Name);
        System.out.println("The model is: " + Model);
        System.out.println("The price of the car is: " + Price);
        System.out.println("The seating capacity of the car is: " +
    Capacity);
        System.out.println("Is it petrol? " + Petrol);
    }
}
```

```

class Main {
    public static void main(String[] args) {
        Car c1 = new Car("Toyota", "Sedan", "5-Seater", 4500, true);
        c1.displayinfo();
    }
}

```

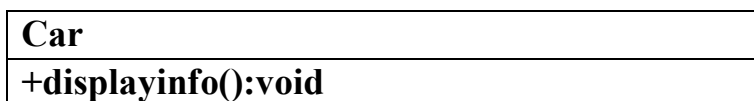
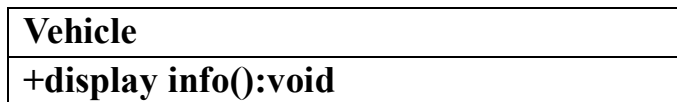
Output:

```

C:\Users\ensar\OneDrive\Desktop\JAVA\week1>java Main
Car name is: Toyota
The model is: Sedan
The price of the car is: 4500
The seating capacity of the car is: 5-Seater
Is it petrol? true

```

CLASS DIAGRAM:



ERROR:

s. no	Expected error	reason
1.	Setting the parameters inside the constructor	We cannot pass the values inside the constructor without setting them first
2.	}	Ending the class and main method is required

2Q) A college is developing automated admission system that verifies students eligibility for UG and PG programs .Each program has different eligibility criteria based on the students percentage in their previous qualification.

- UG admission require minimum 60%
- PG admission require minimum 70%

Program:

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

```
class College {
    String name;
    int qualification;
    int percentage;

    // Constructor
    College(String name, int qualification, int percentage) {
        this.name = name;
        this.qualification = qualification;
        this.percentage = percentage;
    }

    // Default Eligibility method
    public void Eligibility() {
        System.out.println("Name: " + name + ", Qualification: " +
            qualification + ", Percentage: " + percentage);
        System.out.println("The candidate is a fluke");
    }
}

class UG extends College {
    UG(String name, int qualification, int percentage) {
        super(name, qualification, percentage);
    }

    @Override
    public void Eligibility() {
        System.out.println("Name: " + name + ", Qualification: " +
            qualification + ", Percentage: " + percentage);
    }
}
```

```

        System.out.println("The candidate is eligible for UG");
    }
}

class PG extends College {
    PG(String name, int qualification, int percentage) {
        super(name, qualification, percentage);
    }

    @Override
    public void Eligibility() {
        System.out.println("Name: " + name + ", Qualification: " +
            qualification + ", Percentage: " + percentage);
        System.out.println("The candidate is eligible for PG");
    }
}

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

        // Taking inputs
        System.out.println("Enter your name:");
        String name = input.nextLine();

        System.out.println("Enter your qualification (e.g., 12 for high
school, 10 for 10th, etc.):");
        int qualification = input.nextInt();

        System.out.println("Enter your percentage:");
        int percentage = input.nextInt();

        // Close scanner
        input.close();

        // Logic to check eligibility
        College candidate;

        if (percentage >= 70) {
            candidate = new PG(name, qualification, percentage);
        } else if (percentage >= 60) {
            candidate = new UG(name, qualification, percentage);
        } else {

```

```

        candidate = new College(name, qualification, percentage);
    }

    candidate.Eligibility();
}
}

```

Input:

- Name:E. Hima Teja Goud
- Qualification: 12
- Percentage: 95

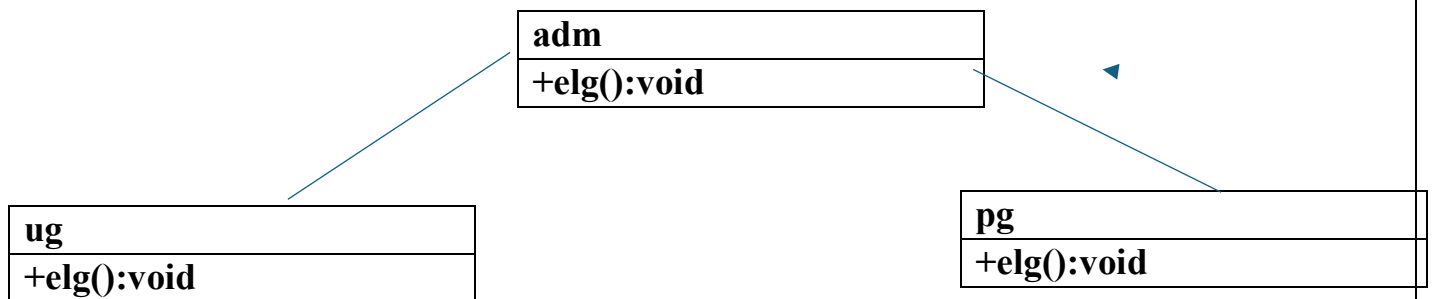
Output:

```

C:\Users\ensar\OneDrive\Desktop\JAVA\week1>java Main
Enter your name:
E. Hima Teja Goud
Enter your qualification (e.g., 12 for high school, 10 for 10th, etc.):
12
Enter your percentage:
95
Name: E. Hima Teja Goud, Qualification: 12, Percentage: 95
The candidate is eligible for PG

```

CLASS DIAGRAM:



ERROR:

	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

3Q)Create a calculator class with overloading methods to perform addition

- Add two doubles
- Add two integer
- Add three integer

Program:

```
public class Calculator{
```

```
    // Method to add two integers
```

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

```
    // Method to add two tuples (represented as arrays)
```

```
    public int add(int[] tuple1, int[] tuple2) {  
        int sum = 0;  
        for (int i = 0; i < tuple1.length; i++) {  
            sum += tuple1[i] + tuple2[i];  
        }  
        return sum;  
    }
```

```
    // Method to add three integers
```

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

```
    public static void main(String[] args) {  
        Calculator calc = new Calculator();
```

```

// Adding two integers
int result1 = calc.add(10, 20);
System.out.println("Addition of two integers: " + result1);

// Adding two tuples (arrays)
int[] tuple1 = {1, 2};
int[] tuple2 = {3, 4};
int result2 = calc.add(tuple1, tuple2);
System.out.println("Addition of two tuples: " + result2);

// Adding three integers
int result3 = calc.add(5, 10, 15);
System.out.println("Addition of three integers: " + result3);
}
}

```

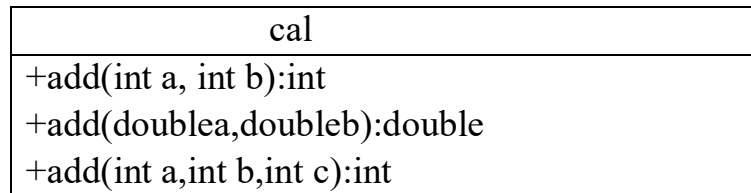
OUTPUT:

```

C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week6>java Calculator
Addition of two integers: 30
Addition of two tuples: 10
Addition of three integers: 30

```

CLASS DIAGRAM:



ERROR:

S	Expected	Reason
.	Error	
N		
o		
.		

1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required

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

CODE:

```

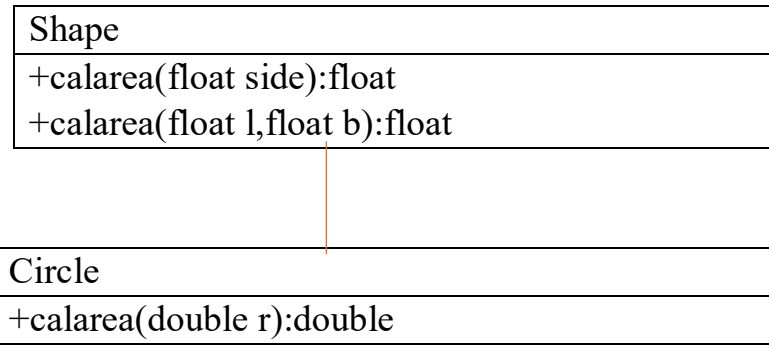
class shape{
public float calarea(float side){ return side*side;
}
public float calarea(float l,float b){ return l*b;
}
}
class circle extends shape{
public double calarea(double r){ return 3.14*r*r;
}
}
class s{
public static void main(String[] args){ circle c=new circle();
System.out.println(c.calarea(10));
}
}

```

OUTPUT:

```
C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week6>java s
100.0
```

CLASS DIAGRAM:



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

WEEK-7

1Q)Write a java program to create an abstract class Animal with an abstract method sound().Create Subclass Tiger and Lion extends the Animal class and implement the sound() method to make a specific sound for each animal

Program:

```
abstract class Animal {

    public abstract void sound();
}

class Lion extends Animal {

    @Override
    public void sound() {
        System.out.println("Lion: Roar!");
    }
}

class Tiger extends Animal {

    @Override
    public void sound() {
        System.out.println("Tiger: Growl!");
    }
}

public class Q1labw7 {
    public static void main(String[] args) {

        Animal lion = new Lion();
        Animal tiger = new Tiger();
    }
}
```

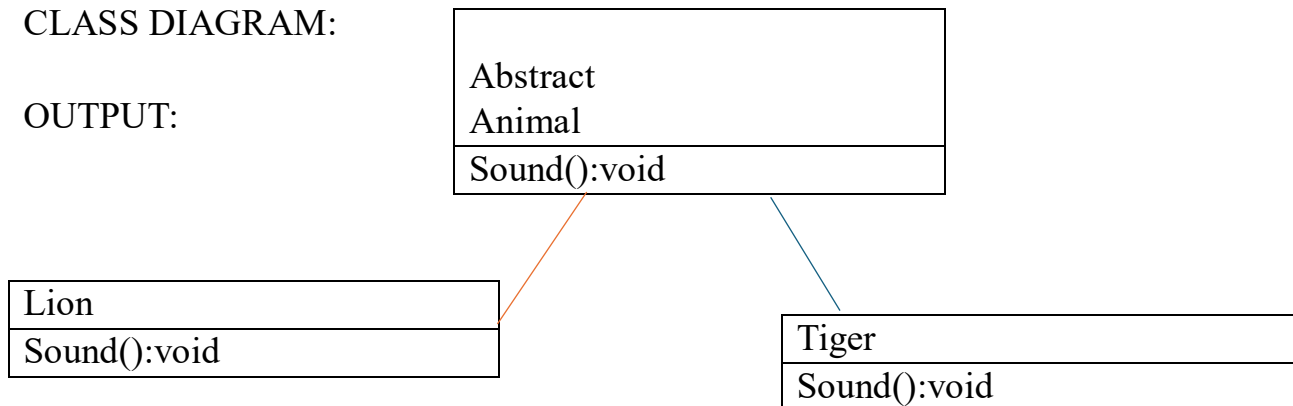
```

lion.sound();
tiger.sound();
}
}

```

CLASS DIAGRAM:

OUTPUT:



```

C:\Users\ensar\OneDrive\Desktop\JAVA\week1>java Q1labw7
Lion: Roar!
Tiger: Growl!

```

ERRORS:

S.NO	errors	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

2Q)Write a java program to create an abstract class Shape3D with an abstract methods Calculate_volume() and Calculate_Surface_area.Create Subclass Sphere and Cube extends the Shape3D class and implement the respective methods to calculate the volume and surface_area of each shape.

Program:

```

abstract class Shape3D {
    public abstract void calculate_volume();
    public abstract void calculate_surf_a();
}

```

```
class Sphere extends Shape3D {
    private double radius;

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

    @Override
    public void calculate_surf_a() {
        double surfaceArea = 4 * Math.PI * Math.pow(radius, 2);
        System.out.printf("Surface Area of Sphere: %.2f%n", surfaceArea);
    }

    @Override
    public void calculate_volume() {
        double volume = (4.0 / 3) * Math.PI * Math.pow(radius, 3);
        System.out.printf("Volume of Sphere: %.2f%n", volume);
    }
}
```

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

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

    @Override
    public void calculate_surf_a() {
        double surfaceArea = 6 * Math.pow(side, 2);
        System.out.printf("Surface Area of Cube: %.2f%n", surfaceArea);
    }

    @Override
    public void calculate_volume() {
        double volume = Math.pow(side, 3);
    }
}
```

```

        System.out.printf("Volume of Cube: %.2f%n", volume);
    }
}

```

```

class Tej {
    public static void main(String[] args) {

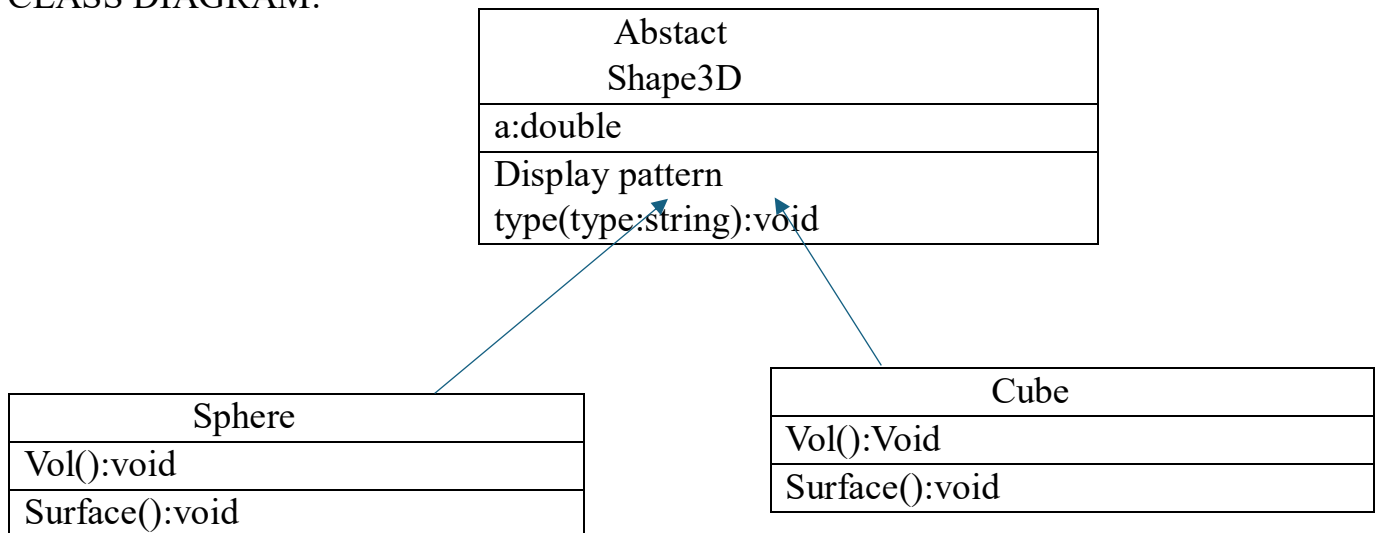
        Shape3D sphere = new Sphere(5);
        Shape3D cube = new Cube(3);

        sphere.calculate_surf_a();
        sphere.calculate_volume();

        cube.calculate_surf_a();
        cube.calculate_volume();
    }
}

```

CLASS DIAGRAM:



OUTPUT:

```
C:\Users\ensar\OneDrive\Desktop\JAVA\week1>java Tej
Surface Area of Sphere: 314.16
Volume of Sphere: 523.60
Surface Area of Cube: 54.00
Volume of Cube: 27.00
```

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

3Q)Write a java program using an abstract class to define a method for pattern printing

-->create an abstract class named patternprinting with an abstract method print pattern (int n) and a concrete method to display the pattern title

-->impletment two sub classes

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 sub classes and print the patterns for a given number of rows

Program:

```
import java.util.Scanner;
abstract class Pattern {

    public abstract void printPattern(int n);
}
```

```

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

```

```

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

```

```

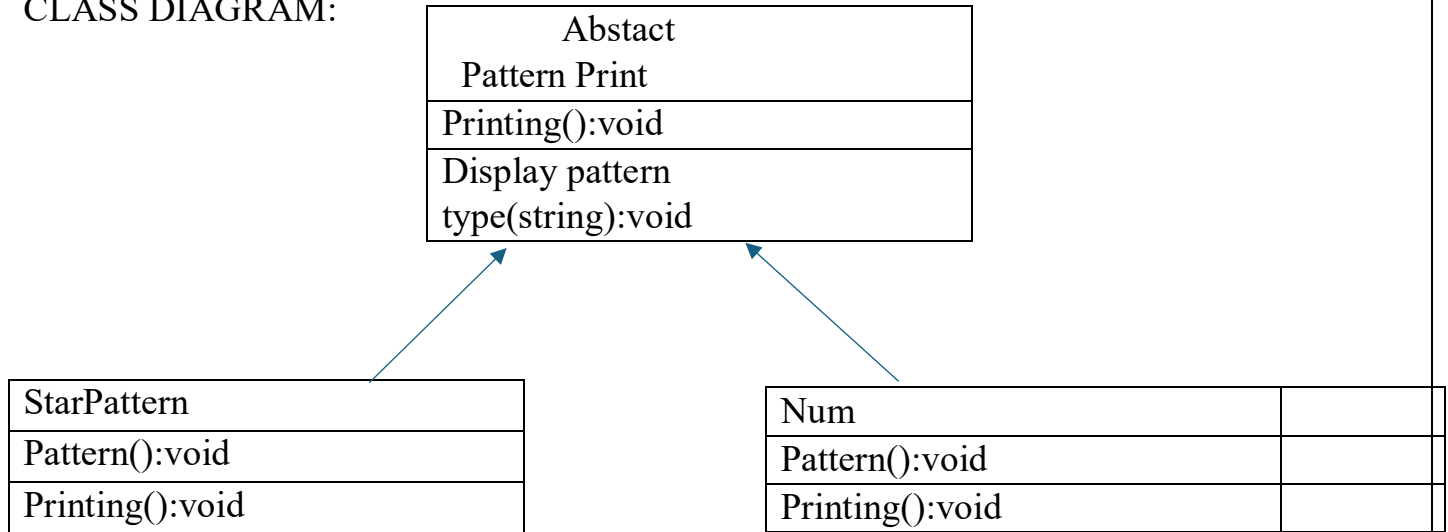
class Hima {
    public static void main(String[] args) {
        Scanner input= new Scanner(System.in);
        System.out.println("enter the n value to select number of rows");
        int n=input.nextInt();

        Pattern rightTriangle = new RightTrianglePattern();
        Pattern numberpattern = new NumberPattern();

        rightTriangle.printPattern(n);
        numberpattern.printPattern(n);
    }
}

```


CLASS DIAGRAM:



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	

OUTPUT:

```
C:\Users\ensar\OneDrive\Desktop\JAVA\week1>java Hima
Enter the value of n (number of rows): 5
Right Triangle Pattern:
*
* *
* * *
* * * *
* * * * *
Number Pattern:
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Week-8

Q)write a java program creating an interface Shape with the get perimeter method create 3 classes rectangle,triangleand circle that implements the shape interface ,implement the getperimeter method for each of the three classes

CODE:

```
import java.util.*;
public interface Shape {
    abstract int getperimeter();
}
class rectangle implements Shape{
    int len,bred;
    rectangle(int len, int bred){
        this.len=len;
        this.bred=bred;}
    public int getperimeter(){
        return 2*(len+bred);
    }
}
class triangle implements Shape{
    int side1,side2,side3;
    triangle(int side1, int side2, int side3){
        this.side1=side1;
        this.side2=side2;
        this.side3=side3;}
    public int getperimeter(){
```

```

        return side1+side2+side3;
    }
}
class Circle implements Shape{
    double radius;
    Circle(int radius){
        this.radius=radius;}
    public int getperimeter(){
        return (int) (2 * Math.PI * radius);
    }
}
class interfaces{
    public static void main(String[] tej){
        Shape Rec= new rectangle(10,5);
        Shape Tri=new triangle(10,10,10);
        Shape circle= new Circle(10);
        System.out.println("rect:"+Rec.getperimeter());
        System.out.println("rect:"+Tri.getperimeter());
        System.out.println("rect:"+ circle.getperimeter());

    }

}

```

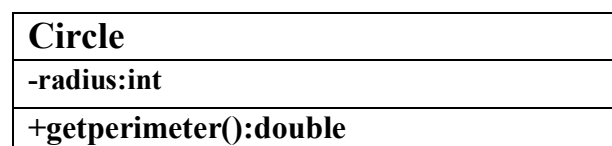
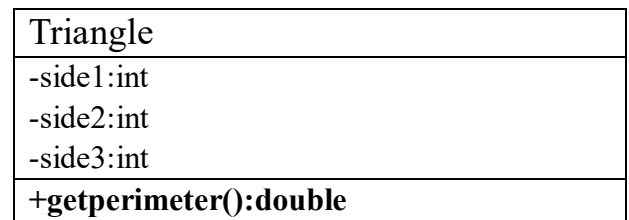
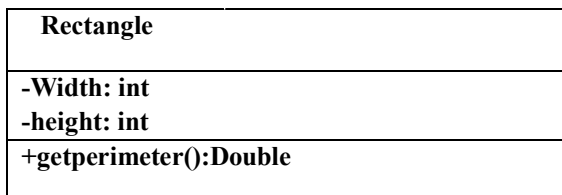
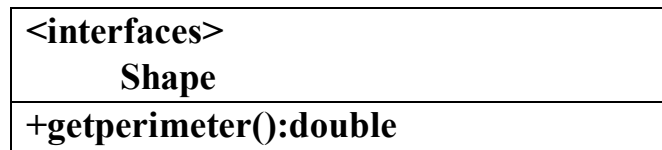
OUTPUT:

```

C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week-8>java interfaces
rect:30
rect:30
rect:62

```

Class Diagrams:



Error Table:

S.NO	Errors	Rectification
1	this.side3=side;	this.side3 = side3;
2	Shape circle = Circle(4);	Shapecircle =newCircle(4);

**2Q)write a java program to create an interface playable with a method play()
That takes no arguments and returns void create three classes football,volleyball and basketball that implements the playable and override the play method to play the respective sports**

CODE:

```
import java.util.*;
interface Playable{
    void Play();
}
class Volleyball implements Playable{
    public void Play(){
        System.out.println("playing vollyball");
    }
}
class Football implements Playable{
    public void Play(){
        System.out.println("football");
    }
}
class Basketball implements Playable {
    public void Play() {
        System.out.println("playing basketball");
    }
}
class SportsGame {
    public static void main(String[] tej){
        Playable Volly= new Volleyball();
        Playable foot= new Football();
    }
}
```

```

    Playable base= new Basketball();
    Volly.Play();
    foot.Play();
    base.Play();
}
}

```

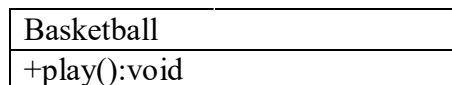
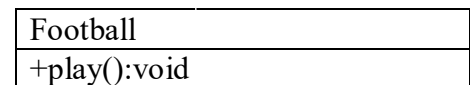
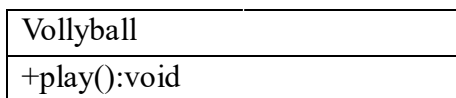
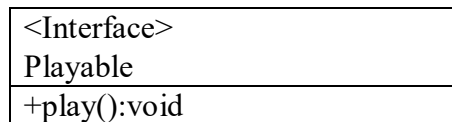
OUTPUT:

```

C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week-8>java SportsGame
playing vollyball
football
playing basketball

```

CLASS DIAGRAM:



Error Table:

s . n o	Expected Error	Reason
1	Settingteparameters inside the constructor	Wecannotpassthevalues inside constructor without setting them first
2	}	Ending the class and main method is required

3Q)write a java program to implement a login system using interfaces

Program:

```
interface LoginSystem {
    boolean login(String id, String password);
}

class UniversityPortal implements LoginSystem {

    @Override
    public boolean login(String id, String password) {
        if (id.equals("student123") && password.equals("pass123")) {
            return true;
        } else {
            System.out.println("Invalid credentials");
            return false;
        }
    }
}

public class Tej {
    public static void main(String[] args) {
        UniversityPortal portal = new UniversityPortal();

        boolean loginSuccess1 = portal.login("student123", "pass123");
        System.out.println("Login successful: " + loginSuccess1);

        boolean loginSuccess2 = portal.login("student123", "wrong pass");
        System.out.println("Login successful: " + loginSuccess2);
    }
}
```

OUTPUT:

```
C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week-8>java Tej
Login successful: true
Invalid credentials
Login successful: false
```

Error Table:

S . n o	Expected Error	Reason
1	Setting the parameters inside the constructor	We cannot pass the values inside the constructor without setting them first
2	}	Ending the class and main method is required

Week-9

1)write a java program to create a method that take integer as parameter and throws an exception if the number is even

Program:

```
public class evennumberchecker {  
    public static void checknumber(int n) throws Exception{  
        if (n%2==0){  
            throw new Exception (" even not allowed");  
        }  
        else {  
            System.out.println("it is an odd number");  
        }  
    }  
    public static void main(String[]Tej){  
        try{  
            checknumber(6);  
        }  
        catch(Exception e){  
            System.out.println("Exception caught "+ e.getMessage());  
        }  
    }  
}
```

OUTPUT:

```
C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week-8>java evennumberchecker  
Exception caught  even not allowed
```

Class diagram:

evennumberchecker
+Checknumber(number:int):void throws exception
+main(args:String args):void

Error Table:

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

2) write a java program to create a method that reads a file and throws an exception if the file is not found

Program:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class Tej {
    public static void main(String[] args) {
        BufferedReader br = null; // Declare BufferedReader outside the try block
        try {
            br = new BufferedReader(new FileReader("E:/Amrita/example.txt"));
            String line; // Corrected 'string' to 'String'
            while ((line = br.readLine()) != null) {
                System.out.println(line); // Print the actual line instead of the string
                "line"
            }
        } catch (IOException e) { // Catch IOException directly
            System.out.println("An error occurred while reading the file: " +
e.getMessage());
        } finally {
            // Ensure the BufferedReader is closed to prevent resource leaks
            if (br != null) {
                try {
                    br.close();
                } catch (IOException e) {
                    System.out.println("Error closing the BufferedReader: " +
e.getMessage());
                }
            }
        }
    }
}
```

OUTPUT:

```
C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week-8>java Tej
An error occurred while reading the file: E:\Amrita\example.txt (The system cannot find the path specified)
```

3Q)write a java program to handle arithmetic exception using try catch and finally

Program:

```
import java.util.Scanner;

public class Week9Q3 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        try {
            System.out.println("Enter first number (numerator):");
            int a = input.nextInt();

            System.out.println("Enter second number (denominator):");
            int b = input.nextInt();

            int result = a / b;
            System.out.println("Result of division: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Error: Cannot divide by zero.");
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        } finally {
            input.close();
        }
    }
}
```

Output:

```
C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week-8>java Week9Q3
Enter first number (numerator):
20
Enter second number (denominator):
4
Result of division: 5
```

Class Diagram:



Error Table:

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

4Q)write a java program to simulate a university system using inner classes

- ✓ **Create an outer class namedd University with a variable UniversityName**
- ✓ **Inside it defgine two non-static in classes**

- 1. Department-With variable like deptName and deptCode and a method to display department details.**
- 2. Student-Variable like stdName and stdCode and a method to display Student details.**
- 3. Create an object for each class and call their methods to display their details and with the university name.**

Program:

```
public class Week9Q4 {  
  
    String universityName = "Amrita University";  
  
    class Department {  
  
        String deptName = "computer science";  
        int deptcode = 101;  
  
        void displayDepartmentInfo() {  
            System.out.println("department" + deptName);  
        }  
    }  
}
```

```

        System.out.println("department" + deptcode);
    }
}

class student {

    String stdname = "Hima Teja Goud";
    int stdcode = 1877;

    void displayStudentInfo() {
        System.out.println("department" + stdname);
        System.out.println("department" + stdcode);
    }
}

public static void main(String[] args) {
    Week9Q4 uni = new Week9Q4();
    System.err.println("University" + uni.universityName);
    System.err.println("Department__Info");
    Week9Q4.Department dept = uni.new Department();
    dept.displayDepartmentInfo();

    System.out.println("***** Student Info *****");
    Week9Q4.student stdent = uni.new student();
    stdent.displayStudentInfo();
}
}

```

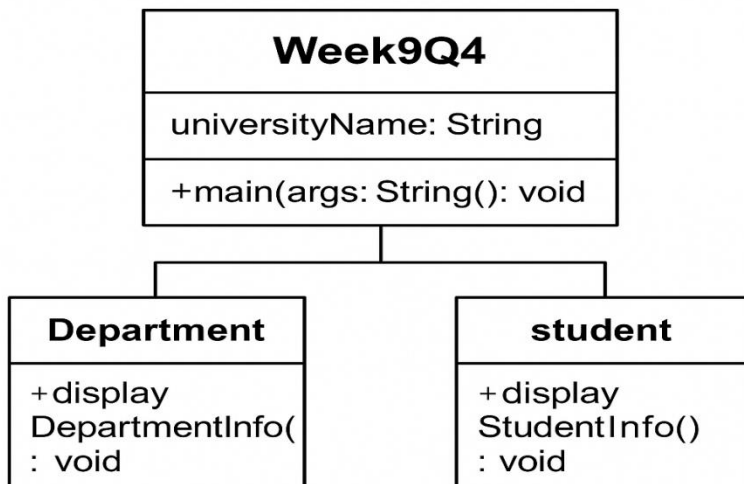
OUTPUT:

```

C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week-8>java Week9Q4
UniversityAmrita University
Department__Info
departmentcomputer science
department101
***** Student Info *****
departmentHima Teja Goud
department1877

```

Class Diagram:



Error Table:

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

Week-10

1Q)Write a java program to generate a password for a student using his/her initials and age. The password displayed should be the string consists of first character of first name, middle name last name with age.

Program:

```
import java.lang.String;
import java.util.Scanner;
public class Q1week10 {
    public static void main(String[] args) {

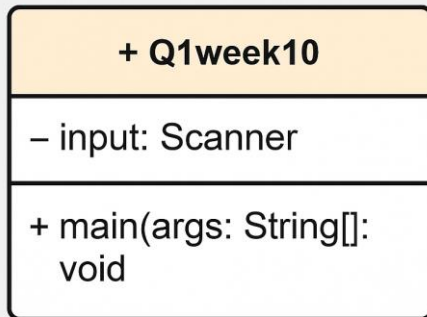
        Scanner input=new Scanner(System.in);

        System.out.println("enter the first name");
        String FN=input.next();
        System.out.println("enter the last name");
        String LN=input.next();
        System.out.println("enter the age");
        int AGE=input.nextInt();
        String initials=FN.substring(0,1)+LN.substring(0,1)+AGE;
        String PIN=initials.toLowerCase();
        System.out.println(initials+":- is the password created");
    }
}
```

Output:

```
C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week-8>java Q1Week10
Enter the first name:
hima teja
Enter the last name:
Enter the age:
19
ht19 :- is the password created
```

Class Diagram



:

2Q)Design and implement a java program that will do the following questions to this string "Welcome! You are practicing Strings concept".

- i) Convert all the alphabets to capital letters and print out the result**
- ii) Convert all alphabets to lower-case letters and print out the result**
- iii) print out the length of the string**
- iv) Print out the index of the concept.**

Program:

```
import java.lang.String;
public class Q2week10 {
    public static void main(String[] args) {
        String Given="welcome! You are practicing strings concept";
        System.out.println("Converting into upper case letters : "+Given.toUpperCase());
        System.out.println("Converting into lower case letters : "+Given.toLowerCase());
        System.out.println("Resulting the length of the string : "+Given.length());
        System.out.println("Finding the index of the given String :"+Given.indexOf("concept"));
    }
}
```

Output:

```
C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week-8>java Q2week10
Converting into upper case letters : WELCOME! YOU ARE PRACTICING STRINGS CONCEPT
Converting into lower case letters : welcome! you are practicing strings concept
Resulting the length of the string : 43
Finding the index of the given String :36
```

Class Diagram:

Q2week10

– Given: String

+ main(args: String():
void

Class Diagram:

3Q) Implement a java program using the below array methods

i) Sorting the elements (numbers and strings) of an array

ii) Convert the array elements into string

iii) Fill the part of an array

iv) Copy the elements of one array into another

Program:

```
import java.util.Arrays;
public class Q3week10 {
    public static void main(String[] args) {
        // 1. Sorting the elements (numbers)
        int[] numbers = {5, 3, 8, 1, 2};
        System.out.println("Original numbers array: " + Arrays.toString(numbers));
        Arrays.sort(numbers);
        System.out.println("Sorted numbers array: " + Arrays.toString(numbers));
        // 1. Sorting the elements (strings)
        String[] strings = {"Banana", "Apple", "Orange", "Mango"};
        System.out.println("Original strings array: " + Arrays.toString(strings));
        Arrays.sort(strings);
        System.out.println("Sorted strings array: " + Arrays.toString(strings));
        // 2. Convert the array elements into strings
        String[] stringArray = Arrays.stream(numbers)
            .mapToObj(String::valueOf)
            .toArray(String[]::new);
        System.out.println("Converted numbers array to strings: " + Arrays.toString(stringArray));
        // 3. Fill part of an array
        int[] filledArray = new int[10];
        Arrays.fill(filledArray, 0, 5, 7); // Fill first 5 elements with 7
        System.out.println("Array after filling part of it: " + Arrays.toString(filledArray));
        // 4. Copy the elements of one array into another
        int[] copiedArray = new int[numbers.length];
        System.arraycopy(numbers, 0, copiedArray, 0, numbers.length);
        System.out.println("Copied array: " + Arrays.toString(copiedArray));
    }
}
```

```
// Close the scanner
```

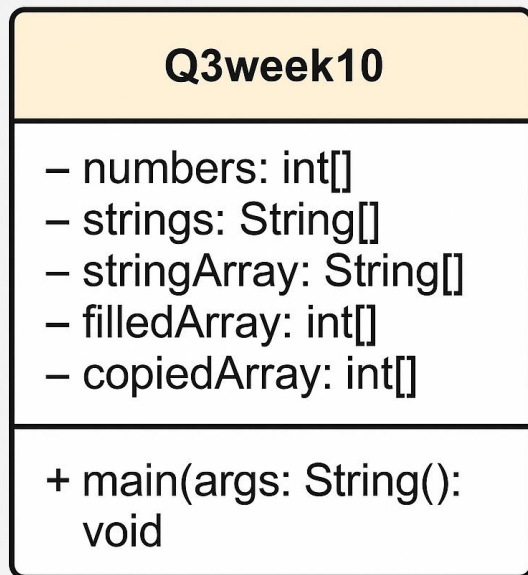
```
}
```

```
}
```

Output:

```
C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week-8>java Q3week10
Original numbers array: [5, 3, 8, 1, 2]
Sorted numbers array: [1, 2, 3, 5, 8]
Original strings array: [Banana, Apple, Orange, Mango]
Sorted strings array: [Apple, Banana, Mango, Orange]
Converted numbers array to strings: [1, 2, 3, 5, 8]
Array after filling part of it: [7, 7, 7, 7, 7, 0, 0, 0, 0, 0]
Copied array: [1, 2, 3, 5, 8]
```

Class Diagram:



Class Diagram:

4) implement a java program using the below Array list

i) Insert an element at particular index in the array list

ii) Modify an element in the array list

iii) access an element from the array list

iv) remove an element from the array list

Program:

```
import java.util.ArrayList;
public class Q4week10 {
    public static void main(String[] args) {
        ArrayList<String> fruits = new ArrayList<>();
        fruits.add("Apple");
        fruits.add("Banana");
        fruits.add("Orange");
        System.out.println("Original ArrayList: " + fruits);
        fruits.add(1, "Mango");
        System.out.println("After inserting 'Mango' at index 1: " + fruits);
    }
}
```



```

fruits.set(2, "Grapes");
System.out.println("After modifying element at index 2: " + fruits);
String fruitAtIndex3 = fruits.get(3);
System.out.println("Element at index 3: " + fruitAtIndex3);
fruits.remove("Banana");
System.out.println("After removing 'Banana': " + fruits);
fruits.remove(0);
System.out.println("After removing element at index 0: " + fruits);
}
}

```

Output:

```

C:\Users\ensar\OneDrive\Desktop\JAVA\week1\week-8>java Q4Week10
Original ArrayList: [Apple, Banana, Orange]
After inserting 'Mango' at index 1: [Apple, Mango, Banana, Orange]
After modifying element at index 2: [Apple, Mango, Grapes, Orange]
Element at index 3: Orange
After removing 'Banana': [Apple, Mango, Grapes, Orange]
After removing element at index 0: [Mango, Grapes, Orange]

```

Class Diagram:

