Object Oriented Programming (23CSE111)

Submitted by	
Name	Sree durga
Roll number	AV.SC.U4CSE24020
Year/sem/section	Year-1/sem-2/CSE-A
Date of submission	

Submitted to	
Name	Dr.B.Raj kumar
Department	CSE
Designation	Asst.Professor

Marks :	

Sno	Programs	Date	Pg 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		03-02-2025		
1.	Write a java program to calculate the area of a rectangle.			
2.a)	Write a program to convert temperature from Celsius to Fahrenheit .			
2.b)	Write a java program to convert temperature from Fahrenheit to Celsius.			
3.	Write a java program to calculate the simple interest			
4.	Write a java program to find the largest of three numbers using ternary operator.			
5.	Write a java program to find the factorial of a number.			
WEEK-3		11-02-2025		
1.	Create a java program with following instructions a)create a class with name car b) Create 4 attributes name car color, car brand, fuel type, milage. c) Create 3 methods named start, stop, services d) Create 3objects named car1, car2, car3. e) Create a constructor which should print “welcome to car garage”			
2.	Write a java program to create a class BackAccount with two methods deposit( ) and withdraw() a) In deposit( ) whenever an amount is deposited it has to be updated with current amount b) In withdraw( ) whenever an amount is withdrawn it has to be less than current amount else print “Insufficient funds”			
WEEK-4		02-03-2025		
1.	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 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.			
2.	To create a java program with class named Myclass with a staticvariable “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			

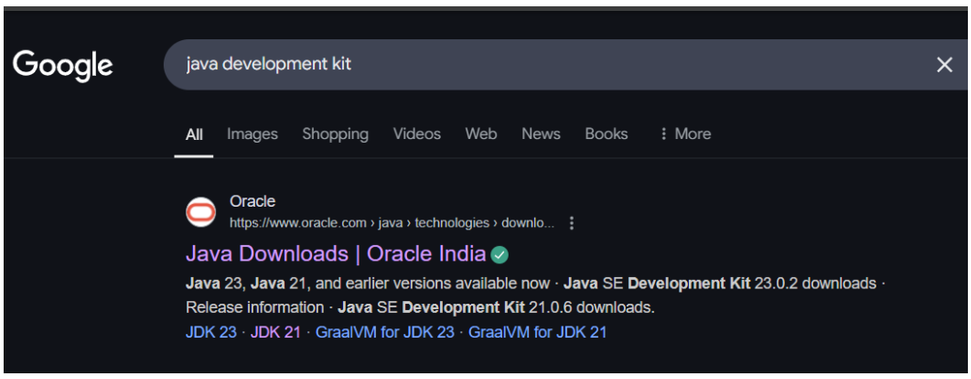
WEEK-5		09-03-2025		
1.	Create a calc using the operations including add, sub, mul, div using multilevel inheritance and display the desired output			
2.	Creating a Rental System			
WEEK-6		16-03-2025		
1.	Write a java program to create a Vehicle class with displayInfo() method , overridden in Car subclass 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 a)additions b)add two integers. c)add two double values d)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 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.			
2.	To 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.			
3.	Write a java program to create a abstract class named patternprinter with an abstract method printpattern(int n) and a concrete method to display the pattern title.			
WEEK-8		21-04-2025		
1.	Write a java program to create an Interface Shape with a getperimeter() method. Create a 3 classes Rectangle,Circle,Triangle and implement the shape interface .Implement the getperimeter() method for each of 3 classes.			
2.	Write a java program to create an Interface playable with a play() method that takes no arugumentsand returns void. Create a 3 classes Football , Basketball,Volleyball and implement the playable			

	interface and override the play() method to play the respective sports.			
3.	Write a java program to implement a login system using interfaces.			
WEEK-9		26-04-2025		
1.	Write a java program that takes integer as parameter and throws an even number exception if the number is even			
2.	Write a java program to create a method that reads a file and thows an exception if the file is not found.			
3.	Write a java program to handle arthematic exception using try, catch and finally.			
4.	Write a java program to stimulate a university system using inner classes Create a outer class named university with a variable universityname Inside it define two non static inner classes.			
WEEK-10				
1.	Write a java program to generate a password for a student using his/her intials and age .The password displayed should the strings 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 this string “Welcome ! you are practicing strings concept” i)Convert all 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 concept			
3.	Implement a java program using the below array method i)Sorting the elements(numbers & strings)of array ii)Convert the array elements into strings iii)Fill the part of any array iv)Copy the elements of one array into another			
4.	Implement a java program using the below array list methods 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 v)Clear the elements from the array list			

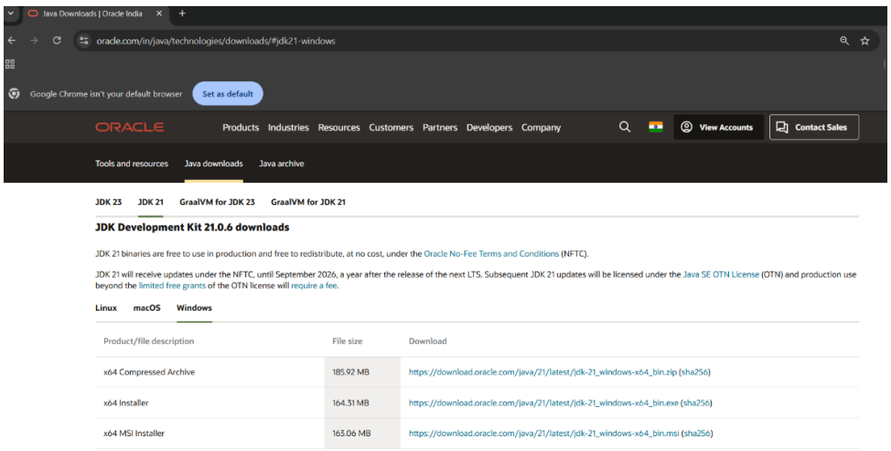
Week1:

1)Aim : Download and installation of java

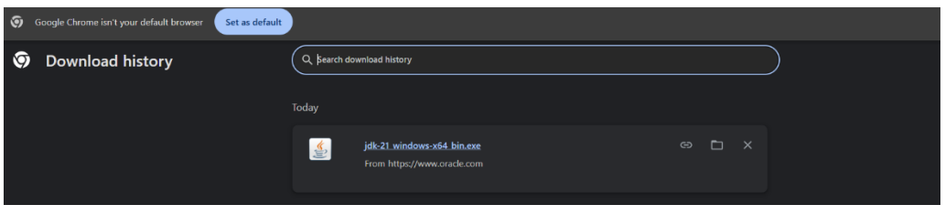
Step 1: Search for java development kit in chrome to download java



Step 2:Open oracle website .Then select JDK21 and download the type of version for your computer



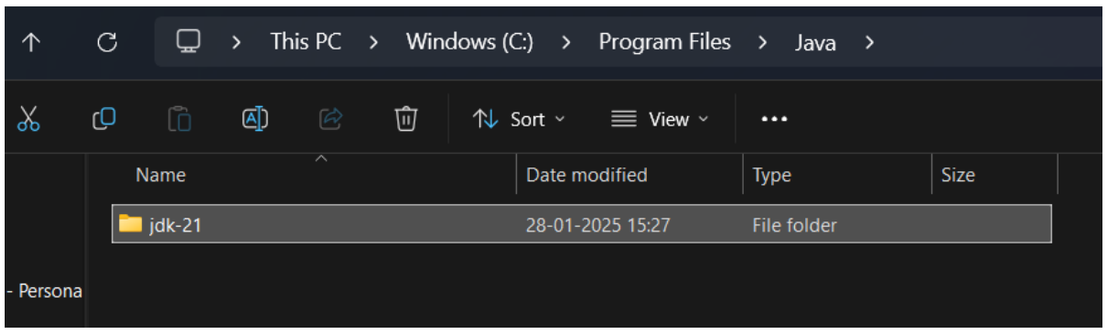
Step 3:After downloading it will appear like the link below. Click on the link for further installation of java software



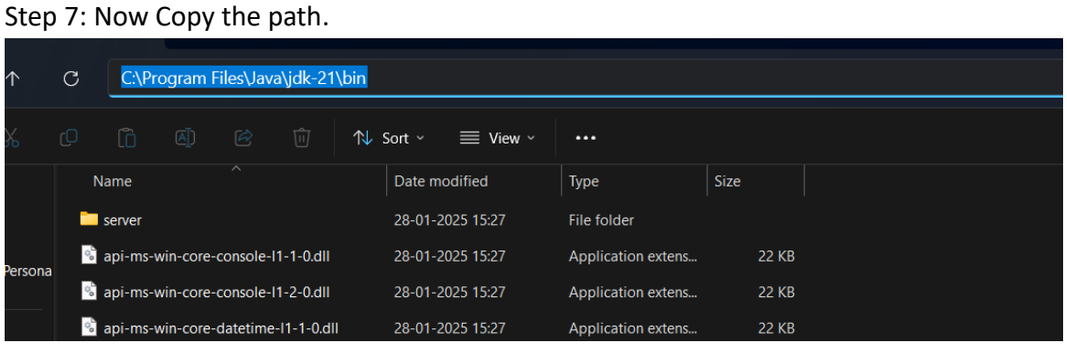
Step 4:Click on next button for further process

Step 5:Click on next button for final installation.

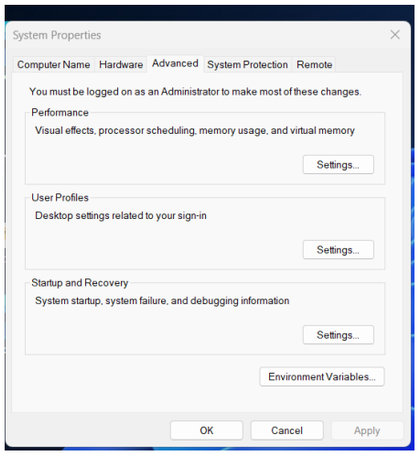
Step 6:To copy the path click on this pc then windows(c:) then program files then java to set path for environment variables



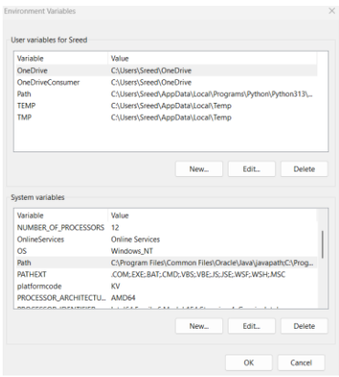
Step 7: Now Copy the path.



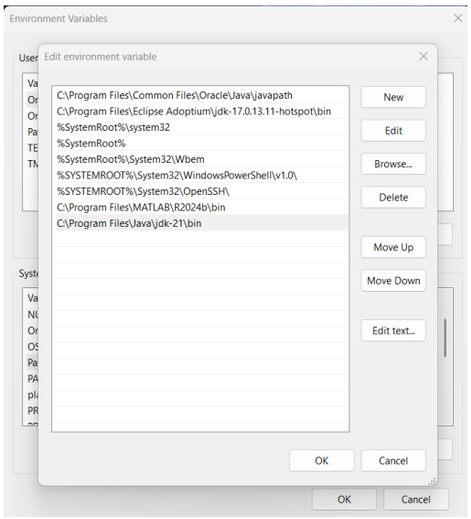
Step 8: Now open environment variables which is displayed below.



Step 9:After clicking environment variables .It will appear like this.



Step 10:To set path click on edit button then paste the copied path here then finally jdk installation is completed.



Step 11:To check the installed version.Open command prompt type java –version then enter. The installed version is displayed.

Program 1

1)Aim: To write a java program to print the message “Welcome to java programming”

Step 1:open notepad<<save the note pad in the path[desktop<<oops<<week 1<<Name.java

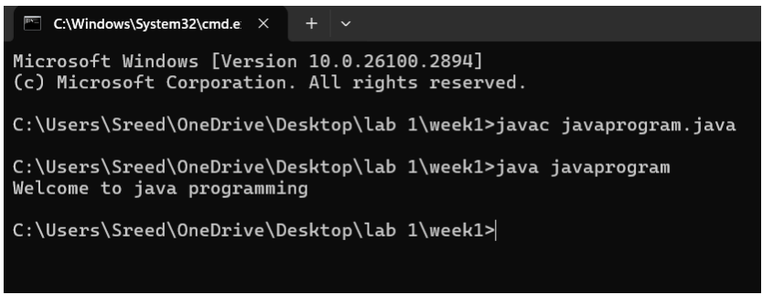
CODE:

Step 2: to the path clear it and type cmd for running the program.

Step 3:Follow commands as: javac Name.java<<enter<<java Name

The program runs successfully.And creates a java class as shown below .

Output:



Step 4:After runner the program system automatically creates a class of it.

ERRORS:

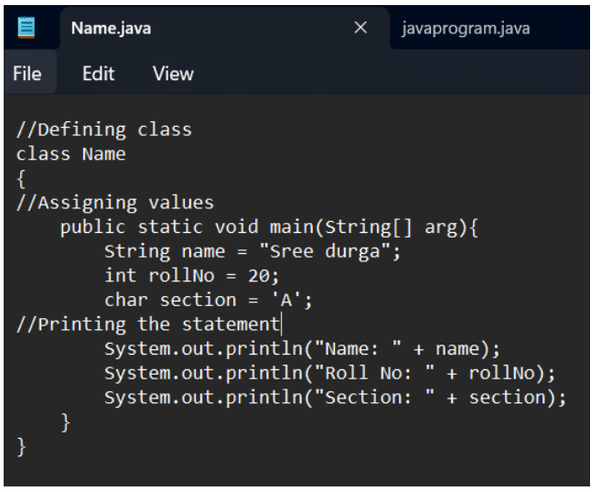
S.NO	ERROR MEASSAGE	ERROR RECTIFICATION
1.	Error:";"expected	Inserted ";"in line 2
2.	Error:"}" expected	Inserted "}" in line 3

Program 2:

Aim: To write a java program that print name,rollno and section of a student

Step 1:open notepad<<save the note pad in the path[desktop<<oops<<week 1<<Name.java

CODE:

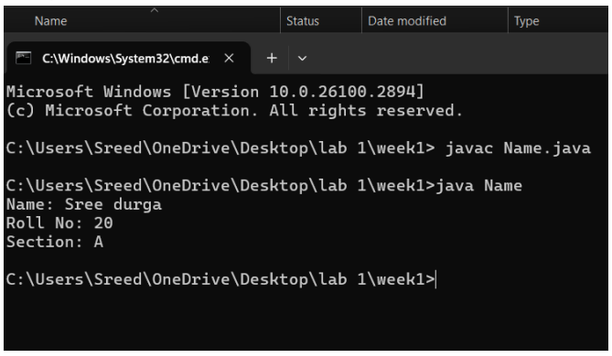


Step 2: to the path clear it and type cmd for running the program.

Step 3:Follow commands as: javac Name.java<<enter<<java Name.

The program runs successfully. And creates a java class as shown below .

Output:



Step 4:After runner the program system automatically creates a class of it.

ERRORS:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error:";"expected	Inserted ";"in line4
2.	Error:"/"unkown sysmbol	Replaced"/"with ":"

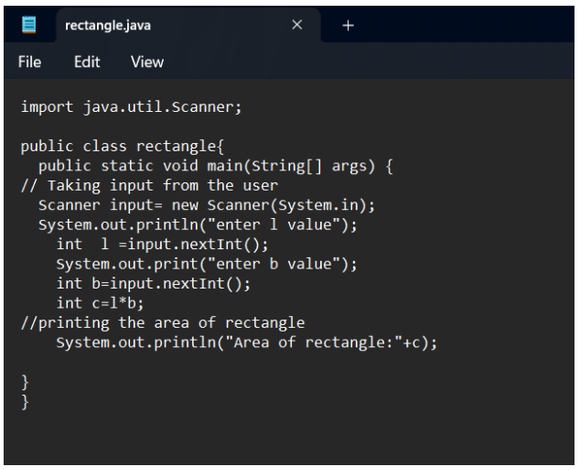
Week-2:

Program 1:

Aim: To write a java program for calculating area of rectangle.

Step 1:open notepad<<save the note pad in the path[desktop<<oops<<week 2<rectangle.java

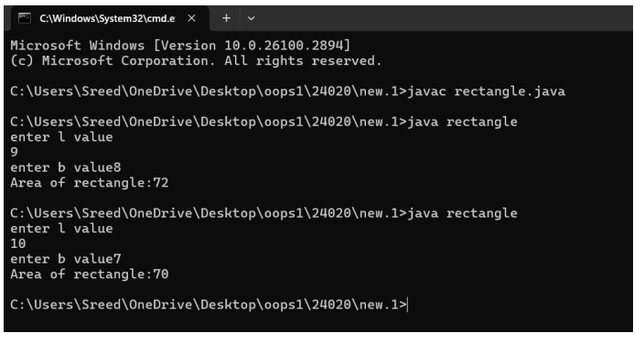
CODE:



Step 2: open code in command prompt and run it.

Step 3: Enter the commands as javac rectangle.java <<java rectangle program is excuted successfully.

Output:



Step 4:after runner the program system automatically creates a class

For it.

ERRORS:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error:”;”expected	Inserted “;”in line7
2.	Error:”?”unkown sysmbol	Replaced”?”with “:”

IMPORTANT POINTS:

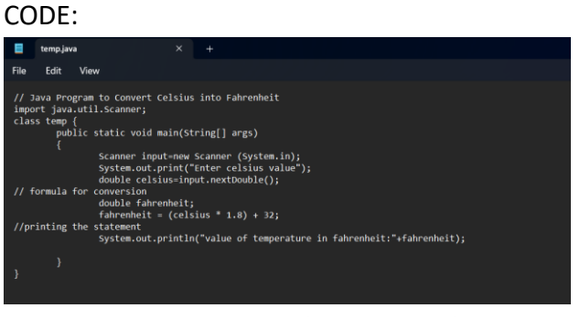
1. Used Scanner library to get input from user in run time .
2. ”import java.util.Scanner;”-step to import library.
3. “Scanner input=new Scanner(System.in);”-step to use the scanner .

Program 2:

A.AIM: To write java program to convert temperature from celcius to farenheit and vice via.

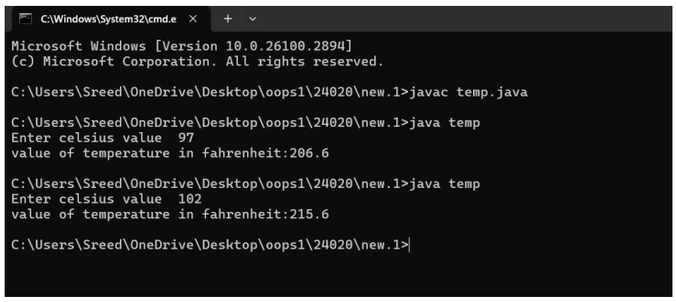
Step 1:open notepad<<save the note pad in the path[desktop<<oops<<week 2<<temp.java

CODE:



Step 2: Open in command prompt and run it.

Step 3: Enter the commands as javac temp<<java temp the program is excuted successfully.



Step 4:After runner the program system automatically creates a class of it.

ERRORS:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error :":" unknown symbol	Replace":" with"="
2.	Error: "scanner"small letter case censitive	Replaced it with "Scanner"

IMPORTANT POINTS:

1.used Scanner library to get input from user in run time.

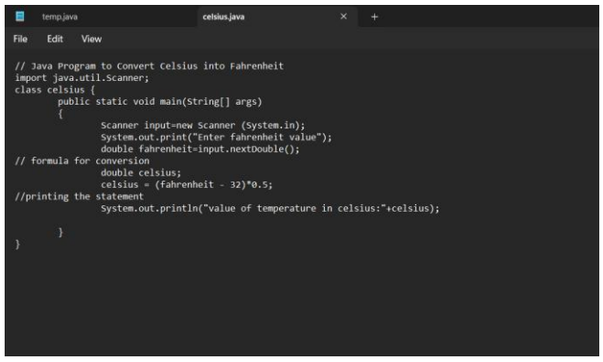
2."import java.util.Scanner;"-step to import library.

3."Scanner input= new Scanner(System.in);"-step to use the scanner.

B.AIM: To write java program to convert temperature from farenheit to celsius.

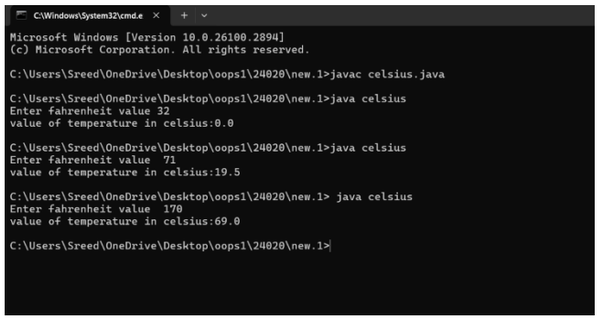
Step 1:open notepad<<save the note pad in the path[desktop<<oops<<week 2<<Celsius.java

CODE:



Step 2:open in command prompt and run it.

Step 3: enter the commands as javac celusis <<java Celsius the program is excuted successfully.



Step 4:After runner the program system automatically creates a class of it.

ERRORS:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error :Celsius file not found	Replaced with Celsius.
2.	Error: } is missing while passing at end of the code	Inserted "}" at the end of the code.

IMPORTANT POINTS:

1.Used Scanner library to get input from user in run time.

2."import java.util.Scanner;"-step to import library.

3."Scanner input= new Scanner(System.in);"-step to use the scanner.

Program 3:

Aim: To write a java program for calculating simple interest.

Step 1:open notepad<<save the note pad in the path[desktop<<oops<<week 2<<simple_interest.java

Code:

simple_interest.java

File Edit View

```
import java.util.Scanner;
class simple_interest{
        public static void main(String[] args){
//Taking input from the scanner
        Scanner input= new Scanner(System.in);
        System.out.println("enter p value");
        int  p =input.nextInt();
        System.out.print("enter t value");
        int t=input.nextInt();
        System.out.print("enter r value");
        int r=input.nextInt();
//Declaring the operation
        int s = p*t*r/100;
//printing the statement
        System.out.println("Simple interest is:"+s);

}
}
```

Step 2:Open in command prompt and run it.

Step 3: enter the commands as javac simple_interest <<java simple interest the program is excuted successfully.

Output:

C:\Windows\System32\cmd.e

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

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>javac simple_interest.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>java simple_interest
enter p value
30
enter t value6
enter r value7
Simple interest is:12

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>java simple_interest
enter p value
70
enter t value3
enter r value1
Simple interest is:2

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>
```

Step 4:After runner the program system automatically creates a class of it.

ERRORS:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error :”T” is not declared	Replaced ”T” with”t”
2.	Error : expected’;’ in line 8	Insert ‘;’ in line 8 end

IMPORTANT POINTS:

1.java is a case sensitive language so “apple” is different from “APPLE”,so clear declaration of variables is important..

2.”Scanner input= new Scanner(System.in);”-step to use the scanner.

Program 4:

Aim: Writing a java program to find biggest among 3 numbers using ternary operator.

Step 1:open notepad<<save the note pad in the path[desktop<<oops<<week 2<<ternary.java

CODE:

ternary.java

File Edit View

```
import java.util.Scanner;

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

        // Input 3 numbers from the user
        System.out.print("Enter first number: ");
        int num1 = input.nextInt();
        System.out.print("Enter second number: ");
        int num2 = input.nextInt();
        System.out.print("Enter third number: ");
        int num3 = input.nextInt();


        int largest = (num1 >= num2 && num1 >= num3) ? num1 :
                      (num2 >= num1 && num2 >= num3) ? num2 : num3;

        // Output the largest number
        System.out.println("The largest number is: " + largest);


    }
}
```

Step 2:Open in command prompt and run it.

Step 3: enter the commands as javac ternary <<java ternary the program is excuted successfully.

Output:

C:\Windows\System32\cmd.e

```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>javac ternary.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>java ternary
Enter first number: 7
Enter second number: 10
Enter third number: 1
The largest number is: 10

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>java ternary
Enter first number: 9
Enter second number: 18
Enter third number: 88
The largest number is: 88

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>
```

Step 4:After runner the program system automatically creates a class of it.

ERRORS:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
	Error :”;” expected in line 6	Insert”;” in end of line 6
	Error :”nextint();” non identified	Replace”next.Int();”

IMPORTANT POINTS:

1.Ternary operators: a shorthand for the if-else statement, used to execute condition-based operations in a single line.

2.It evaluates a Boolean condition and returns trueValue if the condition is true, otherwise it returns falseValue.

Program 5:

Aim: Writing a java program for finding factorial of a number.

Step 1:open notepad<<save the note pad in the path[desktop<<oops<<week 2<<fact.java

CODE:

```
fact.java
File  Edit  View

import java.util.Scanner;

public class fact {

        public static void main(String[] args) {
//Taking input from the user
        Scanner input= new Scanner(System.in);
        System.out.println("enter n value");
        int  n =input.nextInt();
        int i,fact=1;
//Declaring for loop
        for( i = 1; i <= n; ++i)
        {
            // factorial = factorial * i;
            fact *= i;
        }
//Printing the statement
        System.out.printf("Factorial of num is:"+fact);
    }
}
```

Step 2:Open in command prompt and run it.

Step 3: enter the commands as javac fact <<java fact the program is excuted successfully.

Output:

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

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>javac fact.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>java fact
enter n value
7
Factorial of num is:5040
C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>java fact
enter n value
4
Factorial of num is:24
C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>
```

Step 4:After runner the program system automatically creates a class of it.

ERRORS:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
	Error: line-9 illegal start of expression	Rebuilt of the line -9
	Error :iteration error	Correct iteration inserted

IMPORTANT POINTS:

Java for loop is a control flow statement that allows code to be executed repeatedly based on a given condition.

The for loop in java provides an efficient way to iterate over a range of values ,execute code multiple times,or traverse arrays and collections.

Week3:

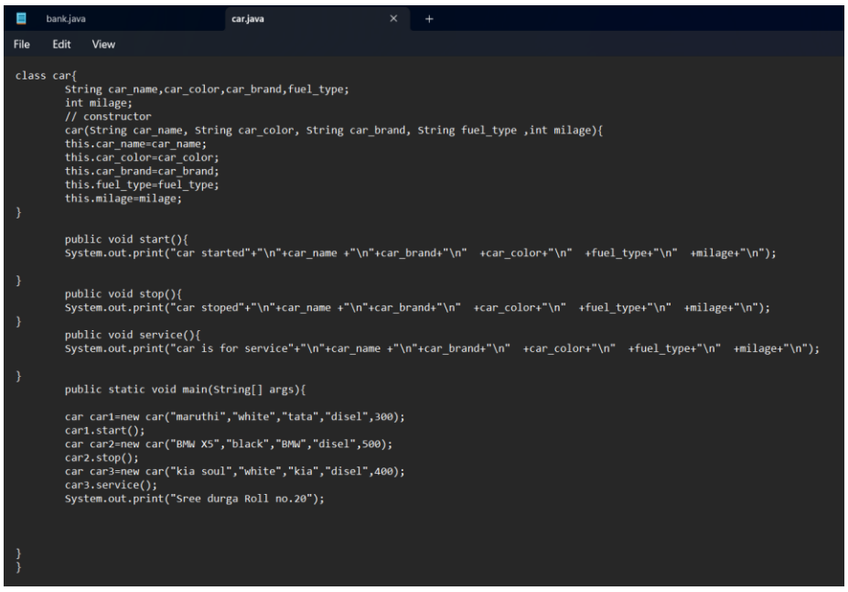
Program 1:

Aim: To write a java program that defines a Car class with attributes, a constructor and methods to perform actions like starting, stopping and servicing a car

Class diagram:

Car
-Name: String -Colour: String -Brand: String -Milage: String
+Car(name,colour,brand milage) +Start(): void +Stop():Void +Service():Void

Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 3<<car.java
Code:



Step 2:Open in command prompt and run it.
Step 3: enter the commands as javac car <<java cars the program is excuted successfully.
Output:

Step 4:After runner the program system automatically creates a class of it.
Errors:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
	Error: ',', ')', or '[' expected	Inserted [
	Error : ; expected in line 6	Inserted ;

IMPORTANT POINTS:
1. Private Variables: Make the car's properties (like make, model, isRunning) private to keep them safe from direct changes.
2. Constructor: Use a constructor to set the car's details (like brand and model) when you create a new car object.

Program 2:

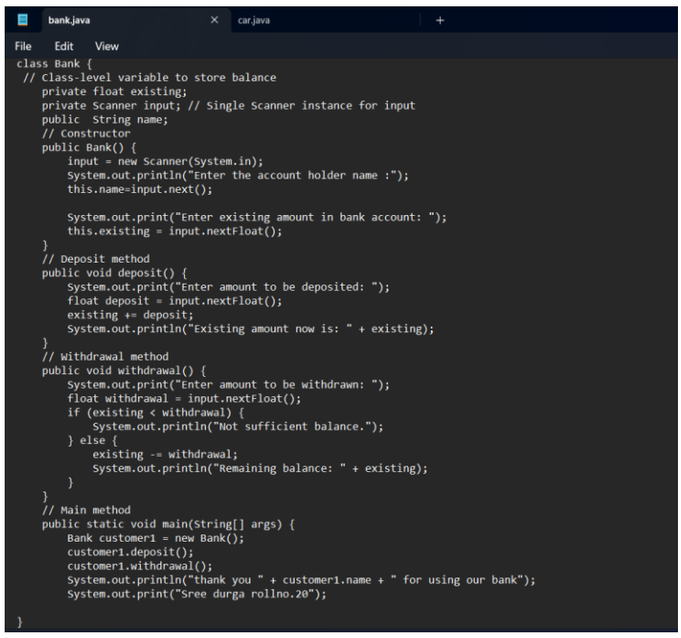
Aim: To write a java program to create a class named bank account with two methods deposit and withdraw.

Class diagram:

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

Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 3<<bank.java

Code:

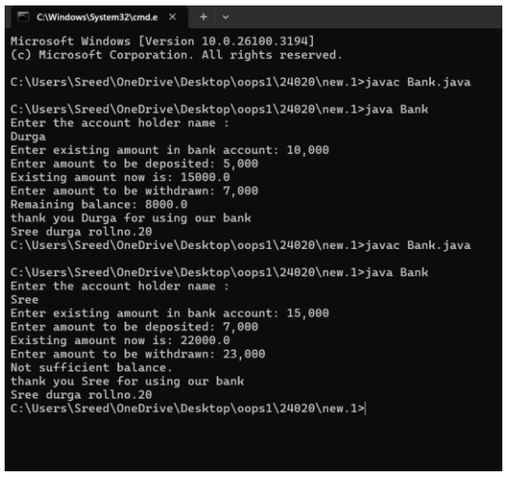


```
bank.java                                car.java
File    Edit    View
class Bank {
 // Class-level variable to store balance
    private float existing;
    private Scanner input; // Single Scanner instance for input
    public  String name;
    // Constructor
    public Bank() {
        input = new Scanner(System.in);
        System.out.println("Enter the account holder name :");
        this.name=input.next();

        System.out.print("Enter existing amount in bank account: ");
        this.existing = input.nextFloat();
    }
    // Deposit method
    public void deposit() {
        System.out.print("Enter amount to be deposited: ");
        float deposit = input.nextFloat();
        existing += deposit;
        System.out.println("Existing amount now is: " + existing);
    }
    // Withdrawal method
    public void withdrawal() {
        System.out.print("Enter amount to be withdrawn: ");
        float withdrawal = input.nextFloat();
        if (existing < withdrawal) {
            System.out.println("Not sufficient balance.");
        } else {
            existing -= withdrawal;
            System.out.println("Remaining balance: " + existing);
        }
    }
    // Main method
    public static void main(String[] args) {
        Bank customer1 = new Bank();
        customer1.deposit();
        customer1.withdrawal();
        System.out.println("thank you " + customer1.name + " for using our bank");
        System.out.print("Sree durga rollno.20");

}
```

Step 2: copy the path then paste it in command prompt and run it.

Step 3: enter the commands as javac bank <<java bank the program is executed successfully.

Output:



```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>javac Bank.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>java Bank
Enter the account holder name :
Durga
Enter existing amount in bank account: 10,000
Enter amount to be deposited: 5,000
Existing amount now is: 15000.0
Enter amount to be withdrawn: 7,000
Remaining balance: 8000.0
thank you Durga for using our bank
Sree durga rollno.20
C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>javac Bank.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>java Bank
Enter the account holder name :
Sree
Enter existing amount in bank account: 15,000
Enter amount to be deposited: 7,000
Existing amount now is: 22000.0
Enter amount to be withdrawn: 23,000
Not sufficient balance.
thank you Sree for using our bank
Sree durga rollno.20
C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1>
```

Step 4: After running the program system automatically creates a class of it.

Errors:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
	Error: illegal start of expression in line 12.	Rebuilt in line 12
	Error : Bank.java:37: error: cannot access bank	Replaced b with B.

Important points:

1)The constructor is called when an object of a class is created. It can be used to set initial values for object attributes.

2)The name of the constructor must match with the name of the class.
3)The constructor doesn’t have a return type.

Week 4:

Program1:

1)Aim: To write a java program with a class named book .The class should contain various attributes such as “title of book” ,” author ”,”year of publication” .It should also contain a constructor with parameters which intalizes “title of book ”,”author ”,”year of publication ”.Create a method which displays the details of the book it is “title of book ”,”author ”,”year of publication ”.Display the details of two books by creating two objects.
Class diagram:

Book
-title: String -author: String Yearofpublished: int
+Book(title,author,yearpublished) +displaydetails():void

Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 4<<book.java
Code:

book.java

File Edit View

```
class book{
        // declaring variables
        String title_of_book,author_name;
        int year_of_publication;
        // constructor
        book(String title_of_book,String author_name,int year_of_publication){
        this.title_of_book=title_of_book;
        this.author_name=author_name;
        this.year_of_publication=year_of_publication;
        System.out.println("Name of the book is"+title_of_book+"Author name"+author_name+"published on"+year_of_publication);
        }
        //method
        public void display(){
        System.out.println();
        }
        //creating object
        public static void main(String[] args){
        book book1=new book("Art of letting go","Ram",2006);
        book1.display();
        book book2=new book("To good to be true","Prajaktha",2020);
        book2.display();
        System.out.println("Sree durga rollno.20");
}
}
```

Step 2: copy the path then paste it in command prompt and run it.
Step 3: enter the commands as javac book.java<<java book the program is executed successfully.

C:\Windows\System32\cmd.e

```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week4>javac book.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week4>java book
Name of the book isArt of letting goAuthor nameRampublished on2006

Name of the book isTo good to be trueAuthor namePrajakthapublished on2020

Sree durga rollno.20

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week4>
```

Step 4: After running the program system automatically creates a class of it.

Errors:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
	Setting the parameters inside the constructor	We cannont pass the values inside constructor without setting them first
	Expected “;” in the line 24	Inserted ; in line 24

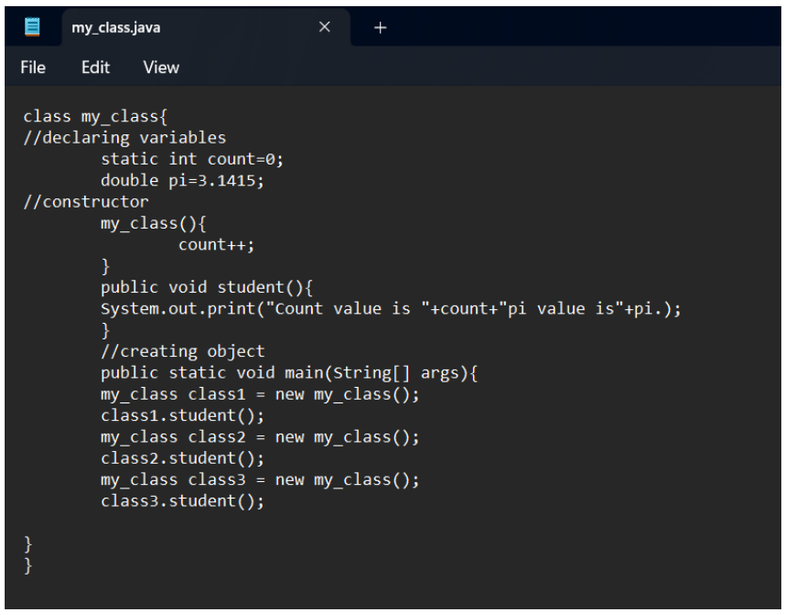
Important points:
1.Declared a ‘static’ variable ‘count’ to keep the track of the number of objects are created static modifier indicates that the variable is a class level variable.
2.Declared a ‘final’ variable to ‘PI’ to represent a constant value ‘final’ modifier indicates that the variables value cannot be changed after it is initialized.

Program 2:

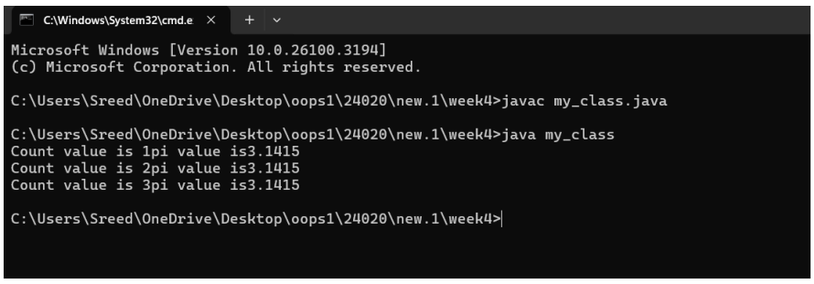
Aim: To write a java program with class named “My class” with a static variable count of int type intalized to zero and a constant variable pi of type double intialized to 3.1415 as attributes of the class. Now define a constructor for my class .That incriments the count variable each time an object is created. Finally print the final values of count and pi variables.
Class diagram:

Myclass
-static count:int Final pi:double=3.1415
+Myclass():void +display():void

Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 4<<My_class.java
Code:



Step 2: copy the path then paste it in command prompt and run it.
Step 3: enter the commands as javac my_class.java<<java my_class the program is executed successfully.



Step 4: After running the program system automatically creates a class of it.

Errors:

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

Important points:

1) Static variables are shared across all instances, while final variables are constants whose values cannot be changed.

2)Constructors are used to initialize objects, and they can modify static variables like count to track object creation.

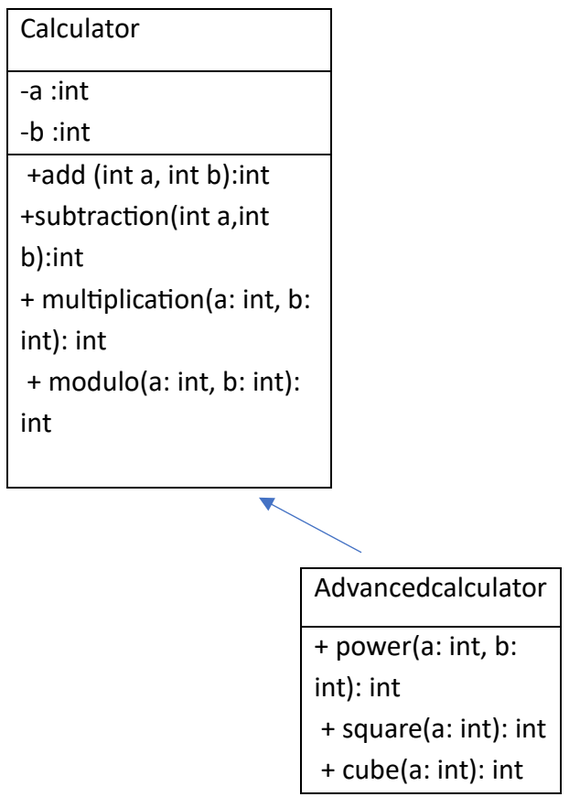
3)Static methods can access static variables directly, but instance methods require an object to access instance variables.

WEEK-5:

Program-1:

Aim :Write a java program to Create a calc using the operations including add, sub, mul, div using multilevel inheritance and display the desired output.

Class diagram:



Step 1: :open notepad<<notepad<<save the note pad in the path[desktop<<oops<<week 5<<Calculatordemo.java

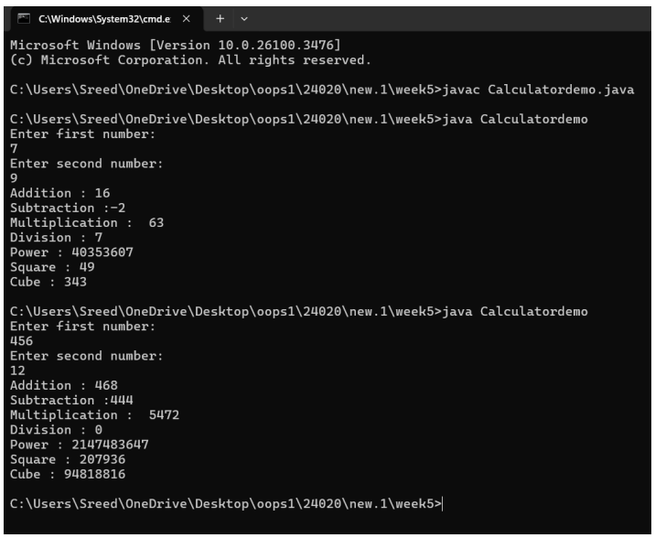
Code:

```
Calculatordemo.java   VehicleRent.java
File  Edit  View

import java.util.Scanner;
//parent class
class Calculator{

        public int add(int a ,int b){
                return a+b;
        }
        public int Subtraction(int a,int b){
                return a-b;
        }
        public int multiplication(int a,int b){
                return a*b;
        }
        public int modulo(int a,int b){
                return a%b;
        }
}
//sub class
class advancedcalculator extends Calculator{
        public int power(int a,int b){
                return (int)Math.pow(a,b);
                }
                public int square(int a){
                        return a*a;
                        }
                        public int cube(int a){
                                return a*a*a;
                                }
                                }
//Main class
public class Calculatordemo{
        public static void main(String[] args) {
        Scanner input=new Scanner(System.in);
        System.out.println("Enter first number:");
        int a =input.nextInt();
        System.out.println("Enter second number:");
        int b =input.nextInt();
                advancedcalculator obj=new advancedcalculator();
                System.out.println("Addition : "+obj.add(a,b));
                System.out.println("Subtraction :"+obj.Subtraction(a,b));
                System.out.println("Multiplication :  "+obj.multiplication(a,b));
                System.out.println("Division : "+obj.modulo(a,b));
                System.out.println("Power : "+obj.power(a,b));
                System.out.println("Square : "+obj.square(((a))));
                System.out.println("Cube : "+obj.cube((a)));
                }
}
```

Step 2: copy the path then paste it in command prompt and run it.

Step 3: enter the commands as javac Calculatordemo.java<<java Calculatordemo the program is executed successfully.

Output:



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

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week5>javac Calculatordemo.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week5>java Calculatordemo
Enter first number:
7
Enter second number:
9
Addition : 16
Subtraction :-2
Multiplication :  63
Division : 7
Power : 40353607
Square : 49
Cube : 343

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week5>java Calculatordemo
Enter first number:
456
Enter second number:
12
Addition : 468
Subtraction :444
Multiplication :  5472
Division : 0
Power : 2147483647
Square : 207936
Cube : 94818816

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week5>
```

Step 4: After running the program system automatically creates a class of it.

Errors:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
	Error: Expected Symbol “;” in line 40	Inserted ;
	Error : Incoorect expression “extend”	Replaced with extend

Important points:

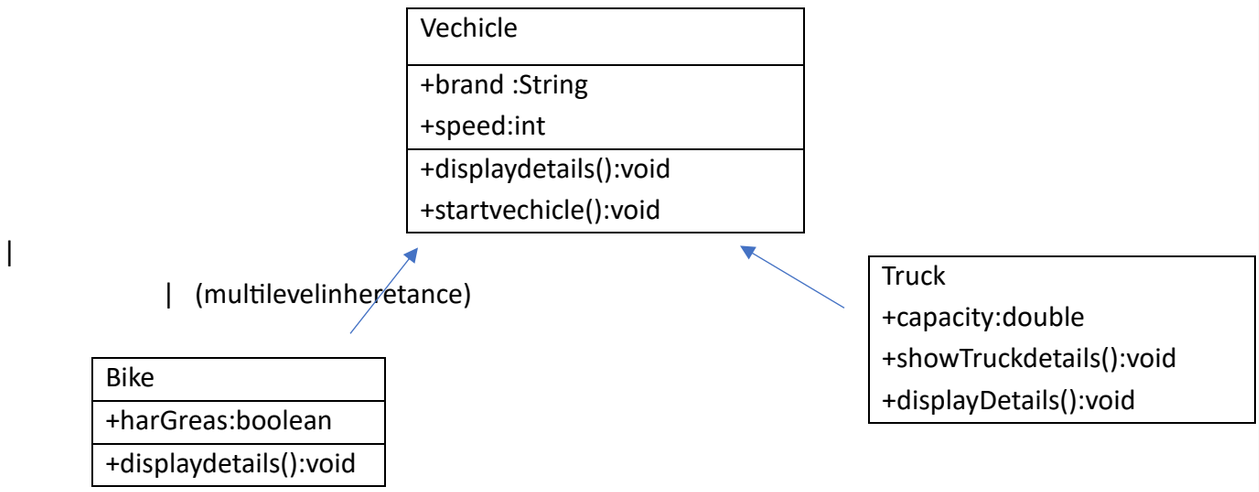
1)Ensure each method in the Calculator and AdvancedCalculator classes has the correct signature. For example, methods should take two integers (or other types depending on the operation) as parameters.

2) The AdvancedCalculator class correctly inherits from the Calculator class, which is a good use of inheritance. This allows you to reuse common functionality like addition, subtraction, etc., while adding specific functionality like power, square, and cube.

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 cars should have an additional properties(attributes)- no.of doors , seating capacity bikes should have a property indicating whether they have gears or not the system should also include a function to display details about each vehicle and indicate when a vehicle is starting each class should have a constructor . a) which oops concept is used in the above program ? Explain why it is useful in this scenario b)If the company decides to add a new type of vehicle truck how would u modify the above program 1) truck should include an additional property called capacity(in tons) 2)create a show truck details method() to display the trucks capacity 3)write a constructor for truck that initializes all the properties c)Implement the truck class and update the main method to create the truck object and also create an object for car and bike subclass. Finally display its details

Class diagram:



Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 5<<VehicleRental.java

Code:

VehicleRent.java

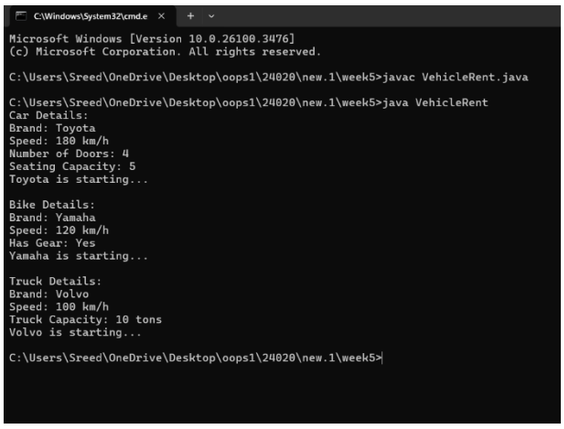
File Edit View

```
class Vehicle {
String brand;
int speed;
Vehicle(String brand, int speed) {
this.brand = brand;
this.speed = speed;
}
void displayDetails() {
System.out.println("Brand: " + brand);
System.out.println("Speed: " + speed + " km/h");
}
void startVehicle() {
System.out.println(brand + " is starting...");
}
}
// Car subclass
class Car extends Vehicle {
int noOfDoors;
30
int seatingCapacity;
Car(String brand, int speed, int noOfDoors, int seatingCapacity) {
super(brand, speed);
this.noOfDoors = noOfDoors;
this.seatingCapacity = seatingCapacity;
}
@Override
void displayDetails() {
super.displayDetails();
System.out.println("Number of Doors: " + noOfDoors);
System.out.println("Seating Capacity: " + seatingCapacity);
}
}
// Bike subclass
class Bike extends Vehicle {
boolean hasGears;
Bike(String brand, int speed, boolean hasGears) {
super(brand, speed);
this.hasGears = hasGears;
}
@Override
void displayDetails() {
super.displayDetails();
System.out.println("Has Gears: " + (hasGears ? "Yes" : "No"));
}
}
// Truck subclass
class Truck extends Vehicle {
double capacity;
Truck(String brand, int speed, double capacity) {
super(brand, speed);
this.capacity = capacity;
}
void showTruckDetails() {
System.out.println("Truck Capacity: " + capacity + " tons");
}
@Override
void displayDetails() {
super.displayDetails();
showTruckDetails();
}
}
// Main class
31
public class VehicleRentalSystem {
public static void main(String[] args) {
Car car = new Car("Toyota", 180, 4, 5);
Bike bike = new Bike("Yamaha", 120, yes);
Truck truck = new Truck("Volvo", 100, 10);
System.out.println("Car Details:");
car.displayDetails();
car.startVehicle();
System.out.println();
System.out.println("Bike Details:");
bike.displayDetails();
bike.startVehicle();
System.out.println();
System.out.println("Truck Details:");
truck.displayDetails();
truck.startVehicle()
}}
```

Step 2: copy the path then paste it in command prompt and run it.

Step 3: enter the commands as javac VehicleRental.java<<java VehicleRental the program is executed successfully.

Output:



Step 4: After running the program system automatically creates a class of it.

Errors:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: Incorrect method call	Inserted ()
2.	Error : Cannot found file name	Replaced Vehiclerental with VechicleRental

Important points:

1) If you have different types of vehicles (e.g., Car, Bike, Truck), you can create subclasses that extend the Vehicle class. This allows you to reuse code and add specific features to each vehicle type.

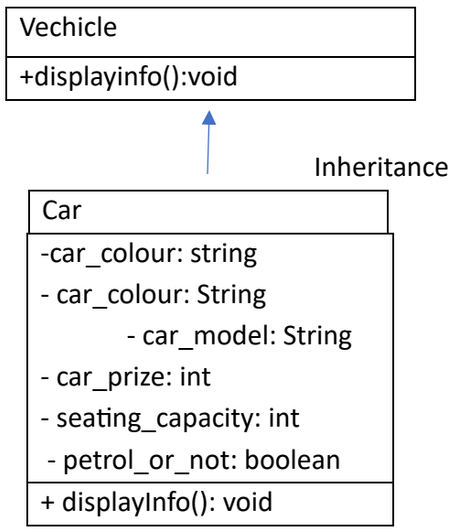
2)You create objects of the Vehicle class or its subclasses to store details about specific vehicles. For example, you might create a Car object with a brand and speed.

Week 6:

Program 1:

Aim: To write a java program to create a vechicle class witha a method displayinfo().override this method in the car subclass to provide specific instance, about car .

Class diagram:



Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 6<<Vehicles.java

Code:

Untitled — Vechicles.java

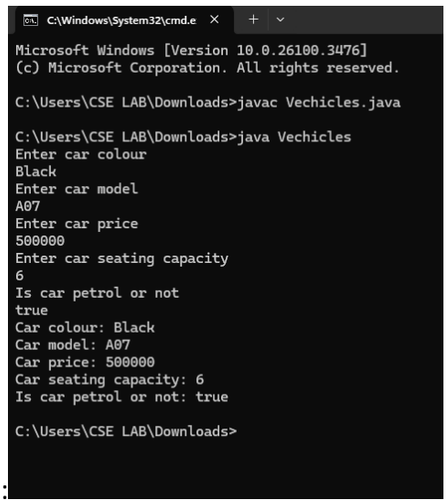
File Edit View

```
import java.util.Scanner;
class Vehicle {
    public void displayInfo() {
        System.out.println("This is a vehicle");
    }
}

class Car extends Vehicle {
    String car_colour, car_model;
    int car_prize, seating_capacity;
    boolean petrol_or_not;

    // Constructor
    Car(String car_colour, String car_model, int car_prize, int seating_capacity, boolean petrol_or_not) {
        this.car_colour = car_colour;
        this.car_model = car_model;
        this.car_prize = car_prize;
        this.seating_capacity = seating_capacity;
        this.petrol_or_not = petrol_or_not;
    }

    // Overriding the displayInfo method
    @Override
    public void displayInfo() {
        System.out.println("Car colour: " + car_colour);
        System.out.println("Car model: " + car_model);
        System.out.println("Car price: " + car_prize);
        System.out.println("Car seating capacity: " + seating_capacity);
        System.out.println("Is car petrol or not: " + petrol_or_not);
    }
}

public class Vechicles{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter car colour");
        String car_colour = sc.nextLine();
        System.out.println("Enter car model");
        String car_model = sc.nextLine();
        System .out.println("Enter car price");
        int car_prize = sc.nextInt();
        System.out.println("Enter car seating capacity");
        int seating_capacity = sc.nextInt();
        System.out.println("Is car petrol or not");
        boolean petrol_or_not = sc.nextBoolean();
        sc.nextLine();
        Car car = new Car(car_colour, car_model, car_prize, seating_capacity, petrol_or_not);
        car.displayInfo();


    }
}
```

Step 2: copy the path then paste it in command prompt and run it.

Step 3: enter the commands as javac Vehicles.java<<java Vehicles the program is executed successfully.

Output:



:

Step 4: After running the program system automatically creates a class of it.

Errors:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
	Error: Missed paranthases	Inserted parantheses
	Error : Undefined symbol in Int	Rectifed by placing int.

Important points:

1)The Car class overrides the displayInfo() method from the Vehicle class. This means that when you call displayInfo() on an object of type Car, it will use the version of the method in Car instead of the version in Vehicle.

2) Polymorphism is demonstrated here, as the displayInfo() method is overridden in the Car class. If you had another class (like Truck or Bike) that also extends Vehicle, you could override displayInfo() in those classes as well. Calling displayInfo() on different objects would then invoke the method specific to the class of the object.

Program 2:

Aim: An automated admission system that verifies student eligibility for UG and PG with different criteria.

.UG requires minimum of 60%

.PG requires minimum of 70%

Class diagram:

Student
-name : String -age : int -marks : double
+checkeligibility(): void

Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 6<<Students.java

Code:

Calculatordemo.java VehicleRent.java **Students.java**

File Edit View

```
class Student {
    String name;
    String email;
    double percentage;

    Student(String name, String email, double percentage) {
        this.name = name;
        this.email = email;
        this.percentage = percentage;
    }

    void getEligibility() {
        System.out.println("Eligibility check not specified for base class.");
    }
}

// Subclass for UG Student
class UGStudent extends Student {

    UGStudent(String name, String email, double percentage) {
        super(name, email, percentage);
    }

    @Override
    void getEligibility() {
        if (percentage >= 60) {
            System.out.println(name + " is eligible for UG admission.");
        } else {
            System.out.println(name + " is NOT eligible for UG admission.");
        }
    }
}

// Subclass for PG Student
class PGStudent extends Student {

    PGStudent(String name, String email, double percentage) {
        super(name, email, percentage);
    }

    @Override
    void getEligibility() {
        if (percentage >= 70) {
            System.out.println(name + " is eligible for PG admission.");
        } else {
            System.out.println(name + " is NOT eligible for PG admission.");
        }
    }
}

// Main
public class Students {
    public static void main(String[] args) {
        UGStudent ug = new UGStudent("Ram", "Ram40@gmail.com", 65.5);
        PGStudent pg = new PGStudent("Nitin", "Nitin10@gmail.com", 68.0);

        ug.getEligibility();
        pg.getEligibility();
    }
}
```

Step 2: copy the path then paste it in command prompt and run it.

Step 3: enter the commands as javac Students.java<<java Students the program is executed successfully.

Output:

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

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week6>javac Students.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week6>java Students
Ram is eligible for UG admission.
Nitin is NOT eligible for PG admission.

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week6>
```

Step 4: After running the program system automatically creates a class of it.

Errors:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: Illegal start of expresiion in line 14	Rebuilt in line 14
2.	Error : Class Name Error	Give the class name correctly

Important points:

1) Handle edge cases like division by zero by checking before performing the operation.

2) Modular Code: Break operations into separate methods for better readability and maintainability.

Program 3:

Aim: To write a java program for creating a calculator class with overloaded methods to perform the give tasks.

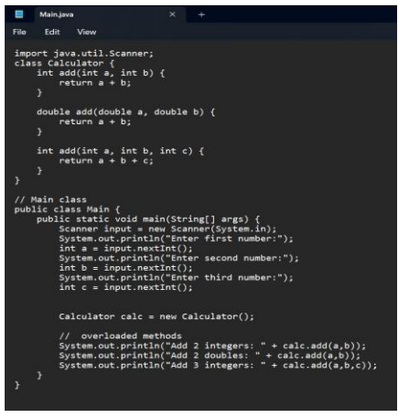
Add 2 integers (2) add 2 doubles (3) add 3 integers

Class diagram:

Calculator
+ add(int,int):int +add (double,double):double +add (int,int,int):int
Main
+main(args: string[]) : void

Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 6<<Main.java

Code:

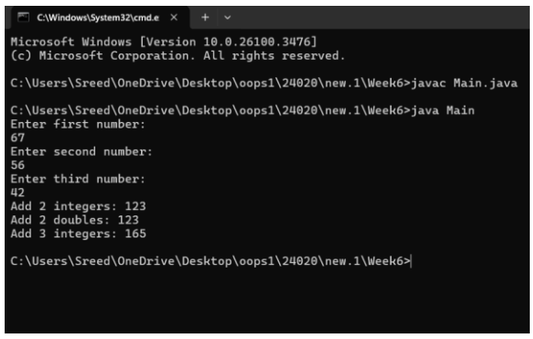


```
Main.java
File  Edit  View

import java.util.Scanner;
class Calculator {
    int add(int a, int b) {
        return a + b;
    }

    double add(double a, double b) {
        return a + b;
    }

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

// Main class
public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter first number:");
        int a = input.nextInt();
        System.out.println("Enter second number:");
        int b = input.nextInt();
        System.out.println("Enter third number:");
        int c = input.nextInt();


        Calculator calc = new Calculator();

        //  overloaded methods
        System.out.println("Add 2 integers: " + calc.add(a,b));
        System.out.println("Add 2 doubles: " + calc.add(a,b));
        System.out.println("Add 3 integers: " + calc.add(a,b,c));
    }
}
```

Step 2: copy the path then paste it in command prompt and run it.

Step 3: enter the commands as javac Main.java<<java Main the program is executed successfully.

Output:



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

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week6>javac Main.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week6>java Main
Enter first number:
67
Enter second number:
56
Enter third number:
42
Add 2 integers: 123
Add 2 doubles: 123
Add 3 integers: 165

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week6>
```

Step 4: After running the program system automatically creates a class of it.

Errors:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: Closing Brackets	Need to Close the brackets
2.	Error : Syntax/ Compilation Error	Absence of Semicolon in line 18

Important points:

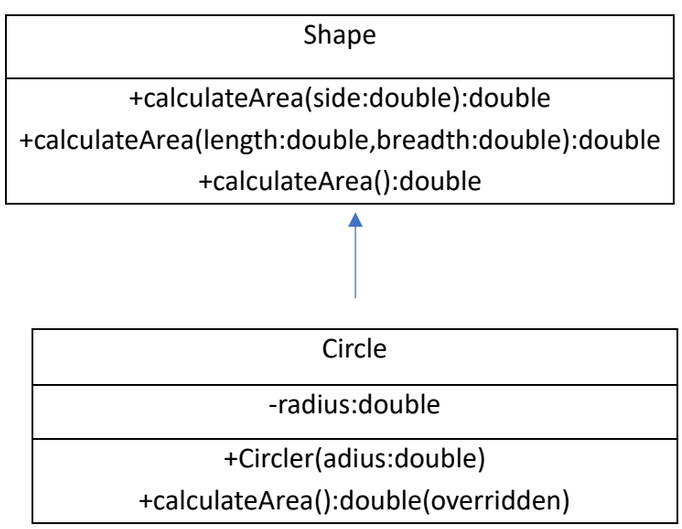
1)We can define multiple methods with the same name but different parameter types (e.g., add(int, int) and add(double, double)).

2) Overloaded methods allow operations on both int and double types.

Program 4:

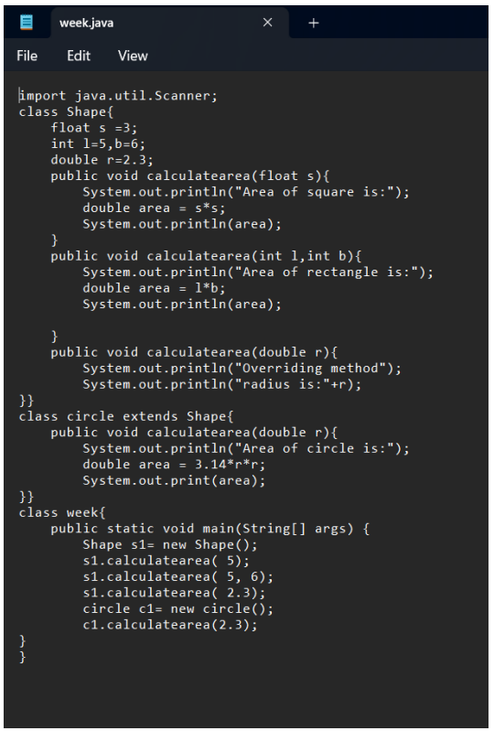
Aim: Create a shape class with a method calculatearea() that is overloaded for different shapes .Then create a subclass circle that overrides the method for circle

CLASS DIAGRAMS:



Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 6<<week.java

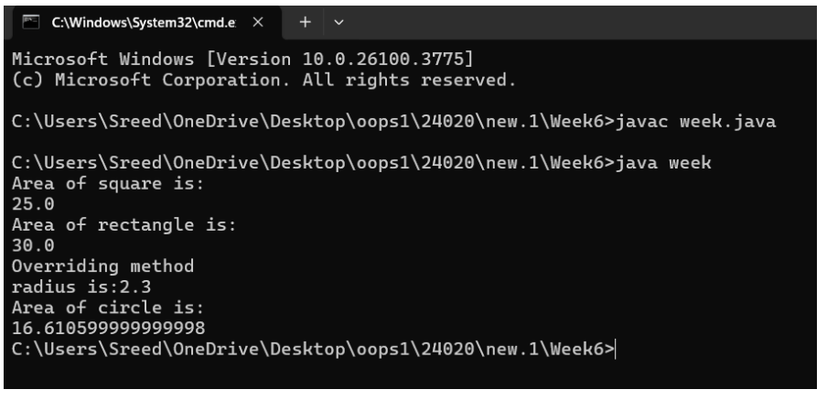
Code:



Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac week.java<<java week the program is executed successfully.

Output:



Step 4: After running the program system automatically creates a class of it.

Errors:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
	Error: }	Ending the class and main method is required.
	Error : Week file not found	Replaced Week with week .

Important points:

1) Subclasses inherit from Shape and override calculateArea() for specific shapes.

2)The same calculateArea() method works differently based on the object type (circle, rectangle, etc.).

3)The base class defines calculateArea() as an abstract method to enforce implementation in subclasses.

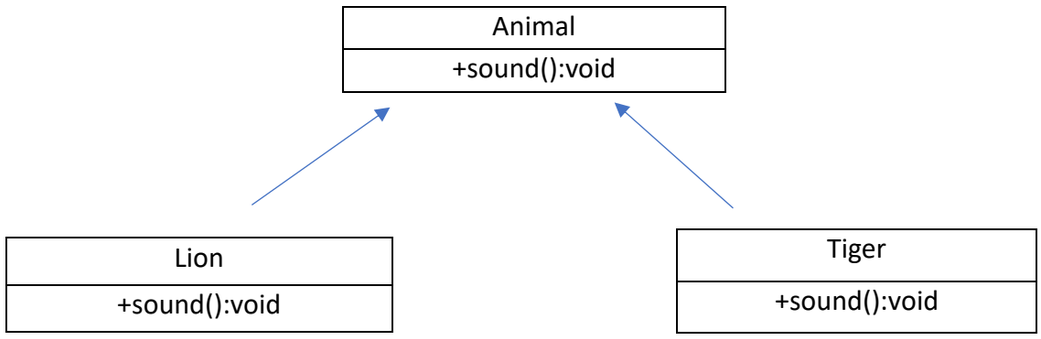
Week 7:

Program-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.

Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 7<<Main.java

Class diagram:



Code:

One.java

File Edit View

```
//Parent class
abstract class Animal{
    abstract void sound();

}
//Sub class 1
class Lion extends Animal{
    void sound(){
        System.out.println("Lion is Roaring");
}
}
//Sub class 2
class Tiger extends Animal{
    void sound(){
        System.out.println("Tiger is growling");
    }
}
//Main class
class Main{
    public static void main(String[] args) {
        Lion lion = new Lion();
        lion.sound();
        Tiger tiger = new Tiger();
        tiger.sound();
    }
}
```

Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac Main.java<<java MAin the program is executed successfully.

Output:

```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week7>javac One.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week7>java One
Lion is Roaring
Tiger is growling

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week7>
```

Step 4: After running the program system automatically creates a class of it.

Errors:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
	Error: Error:java:6line error under “implementation”	Replaced it with “extends ”

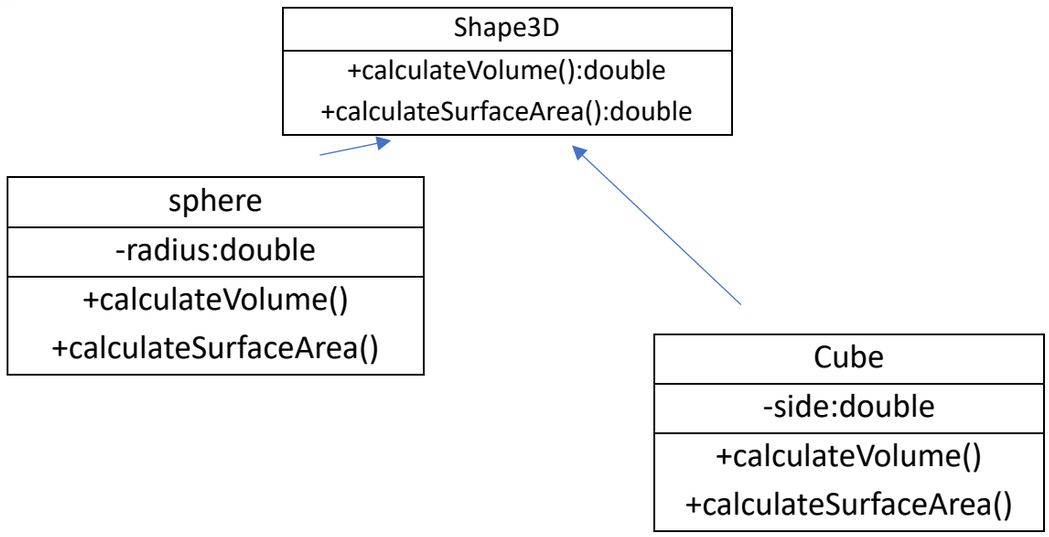
Important points:

1.Cannot be instantiated – Abstract classes cannot create objects directly; they are meant to be inherited by other classes.
2.Contain abstract methods – They can include abstract methods (without implementation) that must be defined in the subclass.
3.Redefining a parent class method – Overriding allows a subclass to provide a specific implementation of a method already defined in its parent class.

Program-2:

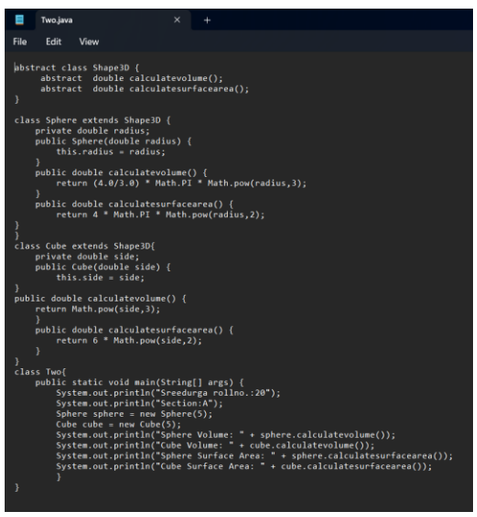
Aim: To write a Java program to create an abstract class Shape3D with abstract methods calculateVolume() and calculateSurfaceArea(). Create subclasses Sphere and Cube that extend the Shape3D class and implement the respective methods to calculate the volume and surface area of each shape.

Class diagram:



Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 7<<Two.java

Code:

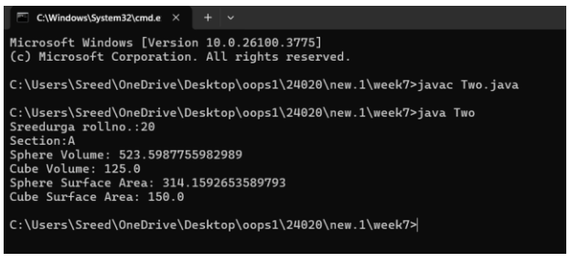


```
abstract class Shape3D {
     abstract  double calculatevolume();
     abstract  double calculatesurfacearea();
}

class Sphere extends Shape3D {
    private double radius;
    public Sphere(double radius) {
        this.radius = radius;
    }
    public double calculatevolume() {
        return (4.0/3.0) * Math.PI * Math.pow(radius,3);
    }
    public double calculatesurfacearea() {
        return 4 * Math.PI * Math.pow(radius,2);
}
}
class Cube extends Shape3D{
    private double side;
    public Cube(double side) {
        this.side = side;
}
public double calculatevolume() {
    return Math.pow(side,3);
    }
    public double calculatesurfacearea() {
        return 6 * Math.pow(side,2);
    }
}
class Two{
    public static void main(String[] args) {
        System.out.println("Sreedurga rollno.:20");
        System.out.println("Section:A");
        Sphere sphere = new Sphere(5);
        Cube cube = new Cube(5);
        System.out.println("Sphere Volume: " + sphere.calculatevolume());
        System.out.println("Cube Volume: " + cube.calculatevolume());
        System.out.println("Sphere Surface Area: " + sphere.calculatesurfacearea());
        System.out.println("Cube Surface Area: " + cube.calculatesurfacearea());
        }
}
```

Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac Main.java<<java Main the program is executed successfully.

Output:



Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: Cannot instantiate abstract class	Rectified it by using subclass: new Sphere(5);
2.	Error : radius ^ 3 ^ is bitwise XOR, not exponent	Rectified it by using Math.pow(radius,3)
3.	Error: abstract calculateVolume();	Rectified it by using Use abstract double calculateVolume();

Important points:

1) An abstract class cannot be instantiated directly.

2)It can contain abstract methods (without a body) and concrete methods (with implementation)

3)Shape-specific properties like radius or side should be private or protected, with appropriate constructors or setters.

Program-3:

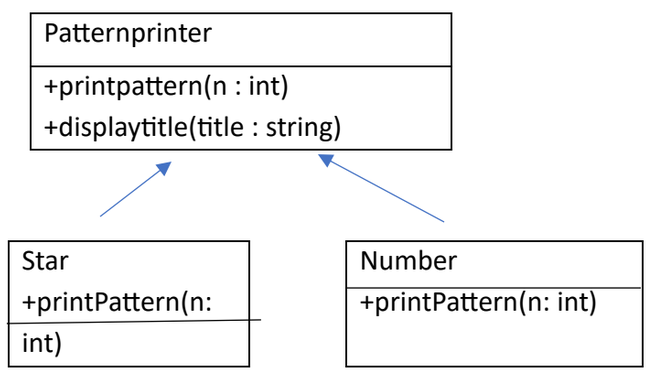
Aim:Write a java program to create a abstract class named patternprinter with an abstract method printpattern(int n) and a concrete method to display the pattern title'

1)Star pattern – prints a rightangled triangle of class(*)

2)Number pattern – prints a rightangled triangle of increasing numbers.

In main method ,create objects of both subclasses and print patterns for a given no.of rows.

Class diagram:



Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 7<<Three.java

Code:

Two.java Three.java

File Edit View

```
abstract class patternprinter{
    abstract void  printpattern(int n);
    abstract void displaytitle();
    }
class Star extends patternprinter{
    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();
                }
                }
                public void displaytitle() {
                    System.out.println("Right angled triangle with stars");
                    }
                    }
class Number extends patternprinter{
    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 void displaytitle() {
                    System.out.println("Right angled triangle with numbers");
                    }
}
public class Three {
    public static void main(String[] args) {
        patternprinter p1 = new Star();
        p1.displaytitle();
        p1.printpattern(7);

        patternprinter p2 = new Number();
        p2.displaytitle();
        p2.printpattern(7);
    }
}
```

Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac Three.java<<java Three the program is executed successfully.

Output:

```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week7>javac Three.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week7>java Three
Right angled triangle with stars
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
Right angled triangle with numbers
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\week7>
```

Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1	Error: PatternPrinter pp = new PatternPrinter();	Instantiate the subclass (e.g., StarPattern, NumberPattern): PatternPrinter pp = new StarPattern();
2	Error : Logical error in the loop.	Rectified it by correcting the loop statement.

Important points:

1)Every subclass of an abstract class must implement all the abstract methods, or the subclass itself will also be abstract.

2)printPattern(int n) is abstract in PatternPrinter, so both StarPattern and NumberPattern must implement it.

3)Even though PatternPrinter is abstract, it can still have concrete methods like displayTitle().

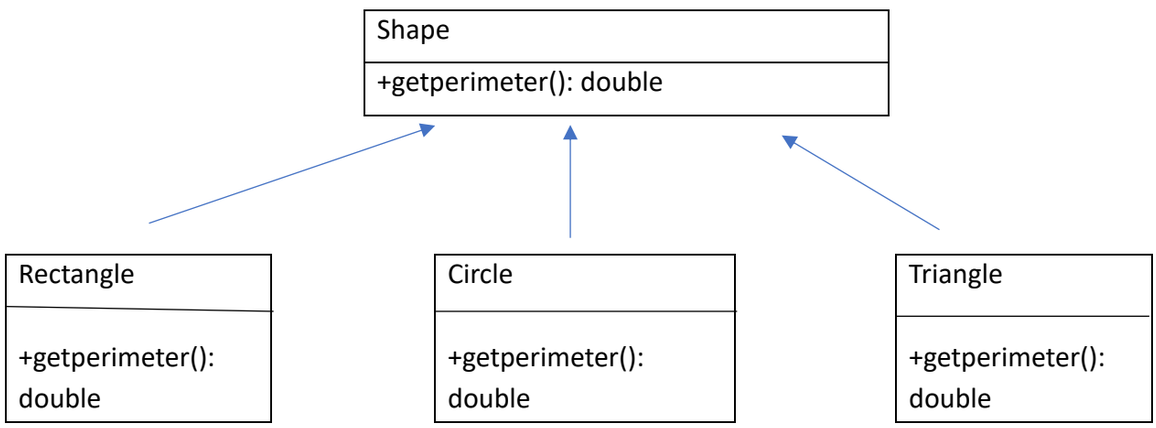
4)The displayTitle() method can be used by any subclass that inherits from PatternPrinter.

Week 8:

Program 1:

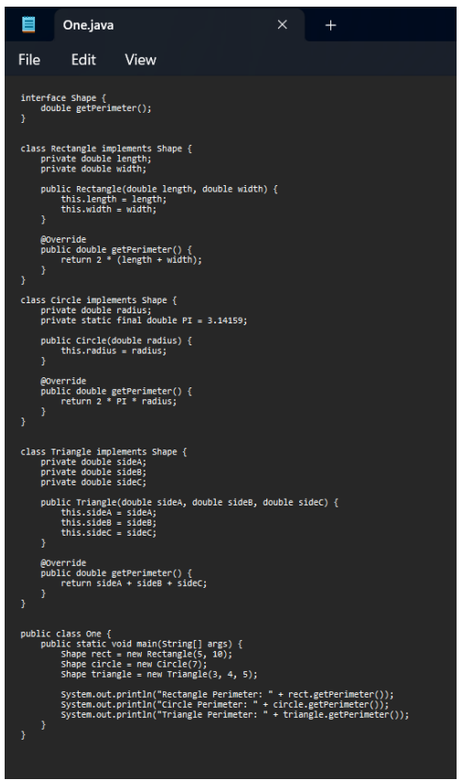
Aim : To write a java program to create an Interface Shape with a getperimeter() method. Create a 3 classes Rectangle,Circle,Triangle and implement the shape interface .Implement the getperimeter() method for each of 3 classes.

Class diagram:



Step 1: :open notepad<<save the note pad in the path[desktop<<oops<<week 8<<One.java

Code:



Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac One.java<<java One the program is executed successfully.

Output:

```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week 8>javac One.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week 8>java One
Rectangle Perimeter: 30.0
Circle Perimeter: 43.98226
Triangle Perimeter: 12.0

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week 8>
```

Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: Negative or Zero Values for Dimensions	Add validation to ensure dimensions are positive.
2.	Error : No Constructor in Interface	Ensure the interface only declares methods and does not have constructors.
3.	Error: Accessing Uninitialized Variables	Ensure all instance variables are initialized via constructors before use.

Important points:

1)Interface Definition: The interface Shape defines the method getPerimeter(), which is implemented by each shape class (Rectangle, Circle, Triangle).

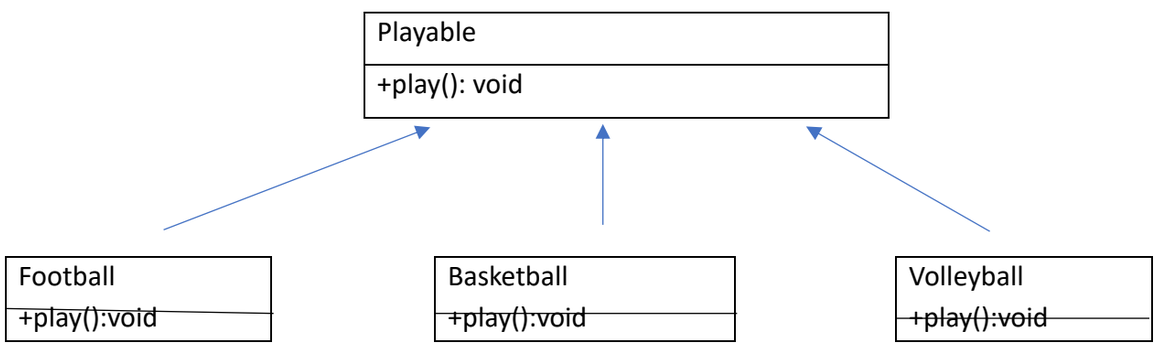
2) Method Overriding: Each class (Rectangle, Circle, Triangle) provides its own implementation of getPerimeter() to calculate the specific perimeter for that shape.

3) Polymorphism: All shapes are treated as Shape objects, allowing you to call getPerimeter() polymorphically without knowing the specific shape.

Program 2:

Aim: To write a java program to create an Interface playable with a play() method that takes no arugumentsand returns void. Create a 3 classes Football , Basketball,Volleyball and implement the playable interface and override the play() method to play the respective sports.

Class diagram:



Step 1: Open notepad<<save the note pad in the path[desktop<<oops<<week 8<<One.java

Code:

Two.java

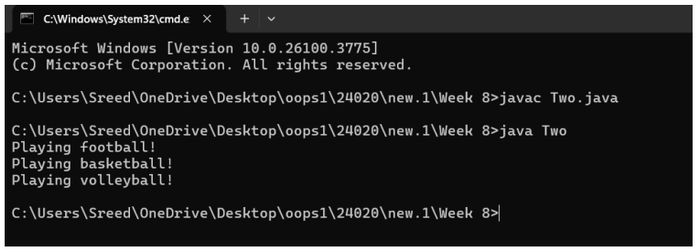
File Edit View

```
interface playable{
    public void play();
}
class Football implements playable{
    public void play() {
        System.out.println("Playing football!");
}}
class Basketball implements playable {
    public void play() {
        System.out.println("Playing basketball!");
    }
}
class Volleyball implements playable {
    public void play() {
        System.out.println("Playing volleyball!");
    }
}
class Two{
    public static void main(String[] args){
        playable game1 = new Football();
        playable game2 = new Basketball();
        playable game3 = new Volleyball();
        game1.play();
        game2.play();
        game3.play();
    }
}
```

Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac Two.java<<java Two the program is executed successfully.

Output:



Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: Missing Method Implementation	Implement the play() method in all classes that implement the interface.
2.	Error : Method Signature Mismatch	Ensure the method matches the interface exactly: void play() (no parameters, no return).

Important points:

1)The play() method takes no arguments and returns void, which means it doesn't require input or return any result. It simply performs an action (simulating the sport).

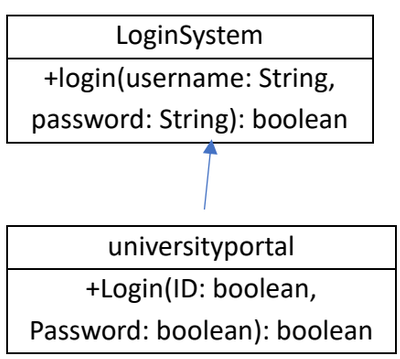
2)The classes encapsulate the logic of playing a particular sport, ensuring that the details are hidden from the main program, making the system modular and easier to extend.

3)You can treat Football, Basketball, and Volleyball objects as Playable references. This allows calling the play() method on any of these objects, enabling flexibility and scalability.

Program 3:

Aim: TO write a java program implement a login system using interfaces.

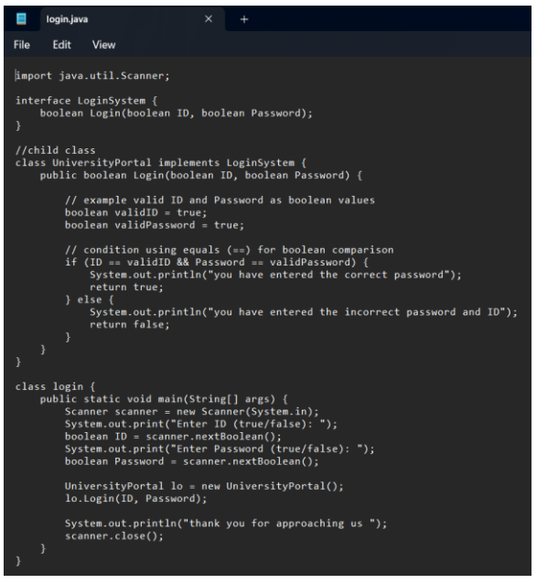
Class diagram:



login
+main(args: String[]): void

Step 1: Open notepad<<save the note pad in the path[desktop<<oops<<week 8<<login.java

Code:

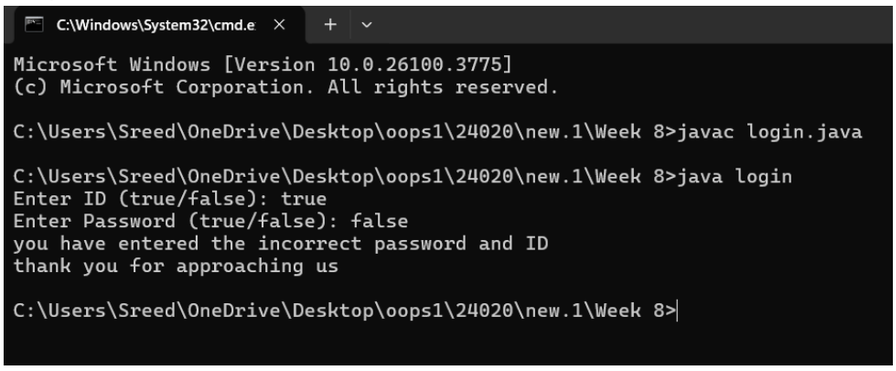


```
import java.util.Scanner;

interface LoginSystem {
    boolean Login(boolean ID, boolean Password);
}

//child class
class UniversityPortal implements LoginSystem {
    public boolean Login(boolean ID, boolean Password) {

        // example valid ID and Password as boolean values
        boolean validID = true;
        boolean validPassword = true;

        // condition using equals (==) for boolean comparison
        if (ID == validID && Password == validPassword) {
            System.out.println("you have entered the correct password");
            return true;
        } else {
            System.out.println("you have entered the incorrect password and ID");
            return false;
        }
    }
}

class login {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter ID (true/false): ");
        boolean ID = scanner.nextBoolean();
        System.out.print("Enter Password (true/false): ");
        boolean Password = scanner.nextBoolean();

        UniversityPortal lo = new UniversityPortal();
        lo.Login(ID, Password);

        System.out.println("thank you for approaching us ");
        scanner.close();
    }
}
```

Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac Two.java<<java Two the program is executed successfully.

Output:



```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week 8>javac login.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week 8>java login
Enter ID (true/false): true
Enter Password (true/false): false
you have entered the incorrect password and ID
thank you for approaching us

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week 8>
```

Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: in decleration of class name	Rectified it by making the first letter capital.
2.	Error: No interface use in main method	Rectifierd it by using interface method.

Important points:

Interface Usage:

You correctly used an interface (LoginSystem) and implemented it in the UniversityPortal class. This is good for abstraction and scalability.

Boolean-based Login:

The logic uses boolean values for ID and Password, which is very simplistic and unrealistic but acceptable for demonstration purposes.

Scanner for Input:

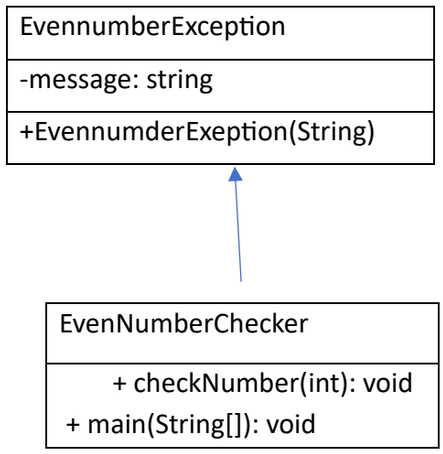
You used Scanner to get input from the user, which is standard for console applications.

WEEK-9:

Program 1:

Aim: To write a java program that takes integer as parameter and throws an even number exception if the number is even.

Class diagram:



Step 1: Open notepad<<save the note pad in the path[desktop<<oops<<week 9<<EvenNumberChecker.java

Code:

EvenNumberChecker.java

File Edit View

```
// Custom exception class
class EvenNumberException extends Exception {
    public EvenNumberException(String message) {
        super(message);
    }
}

public class EvenNumberChecker {

    // Method for even number
    public static void checkNumber(int number) throws EvenNumberException {
        if (number % 2 == 0) {
            throw new EvenNumberException("Even number detected: " + number);
        } else {
            System.out.println("Number is odd: " + number);
        }
    }

    public static void main(String[] args) {
        int testNumber = 8;

        try {
            checkNumber(testNumber);
        } catch (EvenNumberException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}
```

Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac EvenNumberChecker.java<<java EvenNumberChecker the program is executed successfully.

Output:

C:\Windows\System32\cmd.e

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

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week 9>javac EvenNumberChecker.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week 9>java EvenNumberChecker
Exception caught: Even number detected: 8
```

Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: Exception in thread "main" java.lang.NumberFormatException	Rectified it by using try catch
2.	Error:Missing return statement .	Rectified it by placing a return statement.

Important points:

1) Create custom exceptions by extending Exception or RuntimeException.

2)Checked exceptions must be handled with try-catch or declared using throws.

3)Always provide clear exception messages.

4)Validate input to avoid runtime errors.

5)Reuse custom exceptions for similar error types.

Program 2:

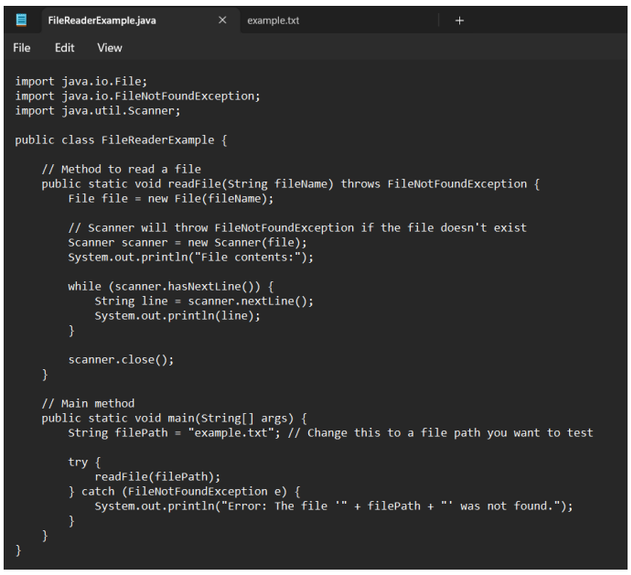
Aim: To write a java program to create a method that reads a file and thows an exception if the file is not found.

Class diagram:

FileReadingExample
+ readFile(String): void + main(String[]): void

Step 1: Open notepad<<save the note pad in the path[desktop<<oops<<week 9<<FileReaderExample.java

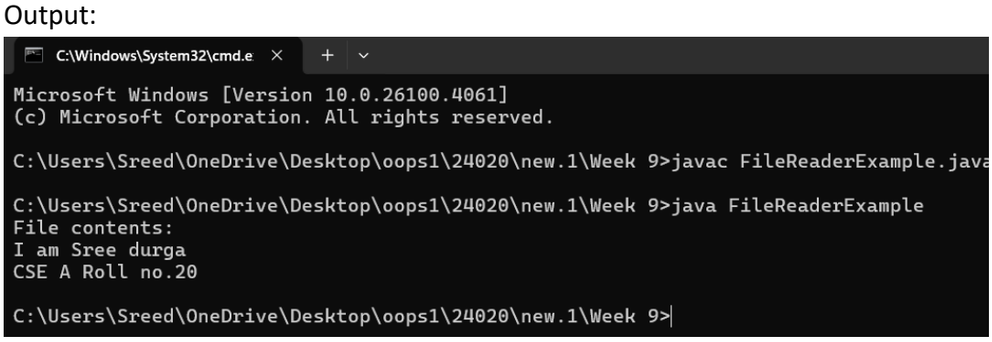
Code:



Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac FileReaderExample.java<<java FileReaderExample the program is executed successfully.

Output:



Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: File reads as empty	Rectified it by verifying file contents manually.
2.	Error: NullPointerException	Rectified it by ensuring the file path is valid and object is initialized.

Important points:

1)Create a File object to represent the file path (e.g., File file = new File("filename.txt");).

2)Scanner is an easy way to read a file line by line.

3)Use a try-catch block or declare throws FileNotFoundException in the method.

4)Before using nextLine(), use scanner.hasNextLine() to avoid errors like NoSuchElementException.

5)Always call scanner.close() to release file resources and prevent memory leaks.

6) In IDEs like IntelliJ or Eclipse, the working directory may be the project root. Ensure your file is in the correct location.

7) Ensure the file has content to avoid unexpected behavior when reading.

Program 3:

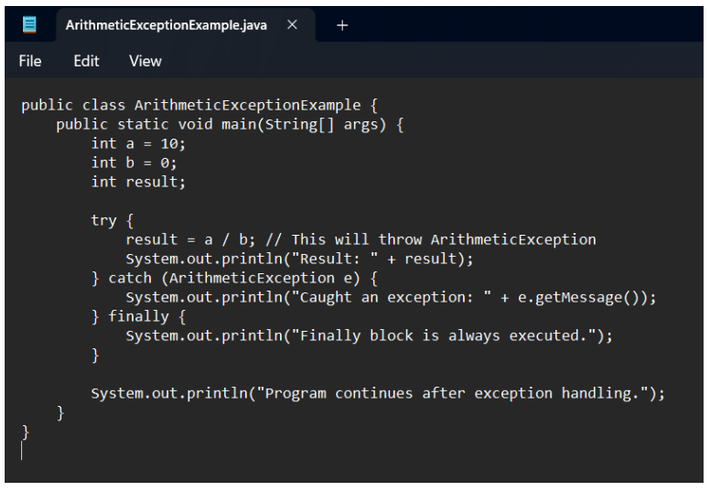
Aim: Write a java program to handle arthematic exception using try, catch and finally.

Class Diagram:

ArithmeticExceptionExample
+main(args: String[]): void
- a: int - b: int - result: int
+try { result = a / b } | +catch(ArithmeticException e) { ... } +finally { ... }

Step 1: Open notepad<<save the note pad in the path[desktop<<oops<<week 9<< ArithmeticExceptionExample.java

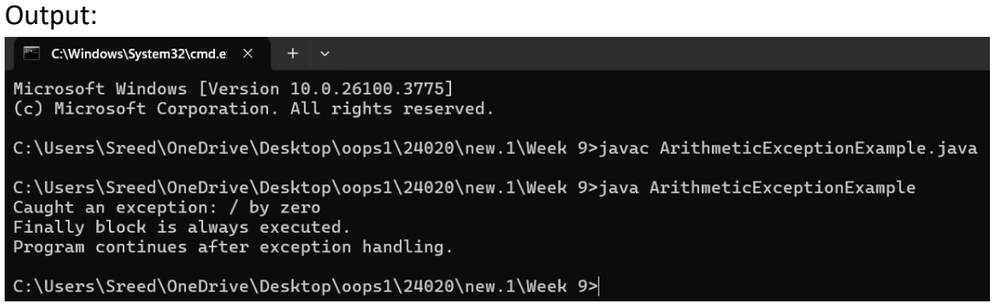
Code:



Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac ArithmeticExceptionExample.java<<java ArithmeticExceptionExample the program is executed successfully.

Output:



Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: ArithmeticException: / by zero	Rectified it by checking the divisor or wrap the code in a try-catch block.
2.	Error:Cannont find the file	Recified it by changing the file name.

Important points:

1)Always use try-catch for risky operations like division, file handling, or array access.

2)Finally always executes, whether an exception is thrown or not.

3)Don’t catch generic Exception unless necessary—prefer specific ones like ArithmeticException, FileNotFoundException, etc.

4)Custom exceptions can be created by extending Exception or RuntimeException.

5) Use multiple catch blocks to handle different exception types.

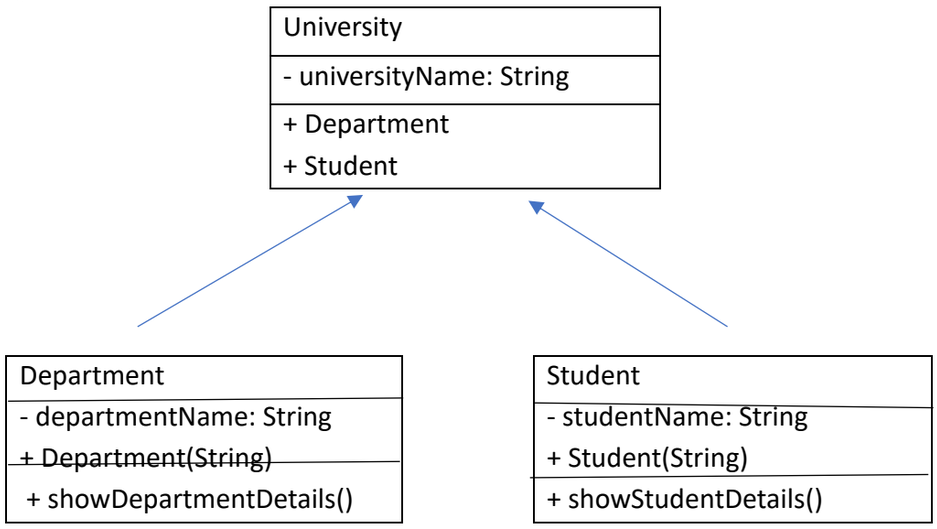
Program 4:

Aim: Write a java program to stimulate a university system using inner classes

a)Create a outer class named university with a variable universityname

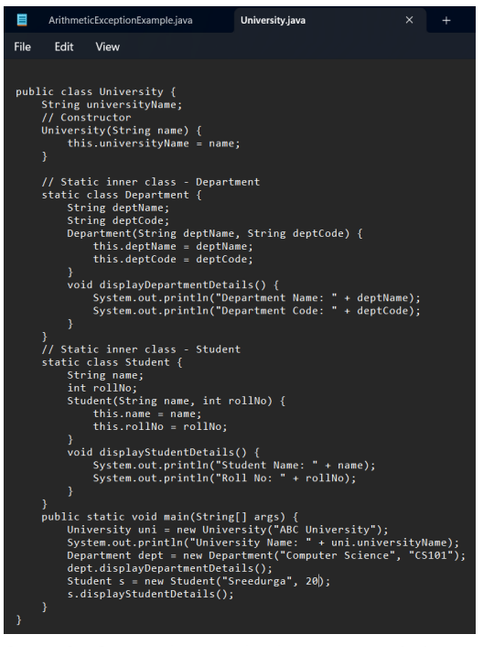
b)Inside it define two non static inner classes.

Class diagram:



Step 1: Open notepad<<save the note pad in the path[desktop<<oops<<week 9<< University.java .

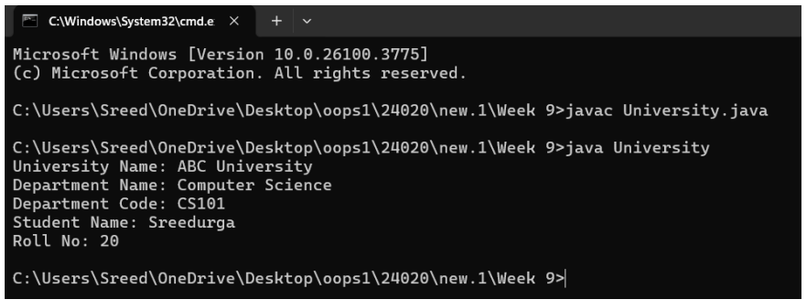
Code:



Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac University.java<<java University the program is executed successfully.

Output:



Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: Cannot make a static reference to the non-static inner class	Inner classes require an instance of the outer class.
2.	Error: Class Department is not visible	Rectified it by ensuring inner classes are public if accessed outside the outer class.

Important points:

1)Non-static inner classes require an outer class instance to be instantiated.

2)Inner classes can access all members of the outer class, including private ones.

3)Inner classes are useful for logically grouping classes used only within the outer class.

4)Inner classes cannot have static declarations (except for static final constants).

5)You can define multiple inner classes inside one outer class.

6)Inner classes help in maintaining encapsulation and making code modular.

Week-10:

Program-1:

Aim: Write a java program to generate a password for a student using his/her intials and age .The password displayed should the strings consists of first character , of first name, middle name, last name with age.

Class diagram:

StudentPassword
- firstName: String - middleName: String - lastName: String - age: int
+generatepassword(): String

Step 1: Open notepad<<save the note pad in the path[desktop<<oops<<week 10<< StudentPassword.java.

Code:

StudentPassword.java

File Edit View

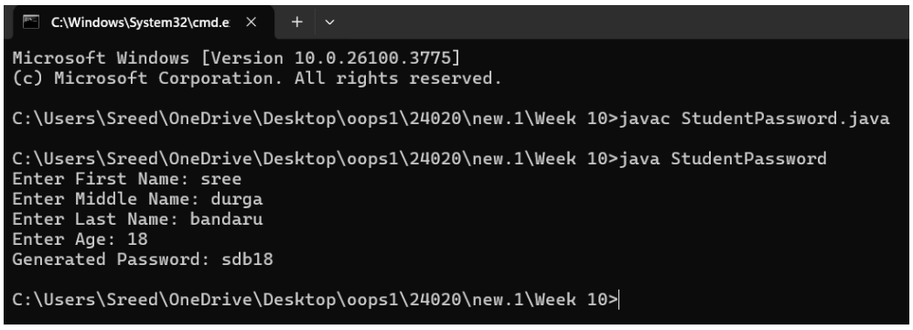
```
import java.util.Scanner;

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

        // Getting user input
        System.out.print("Enter First Name: ");
        String firstName = input.nextLine().trim();

        System.out.print("Enter Middle Name: ");
        String middleName = input.nextLine().trim();

        System.out.print("Enter Last Name: ");
        String lastName = input.nextLine().trim();

        System.out.print("Enter Age: ");
        int age = input.nextInt();

        // Generating initials
        char firstInitial = firstName.isEmpty() ? '_' : firstName.charAt(0);
        char middleInitial = middleName.isEmpty() ? '_' : middleName.charAt(0);
        char lastInitial = lastName.isEmpty() ? '_' : lastName.charAt(0);

        // Creating the password
        String password = "" + firstInitial + middleInitial + lastInitial + age;

        // Display the generated password
        System.out.println("Generated Password: " + password);
    }
}
```

Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac StudentPassword.java.

<<java StudentPassword the program is executed successfully.

Output:



Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: StringIndexOutOfBoundsException	Rectified it by ensuring string is not empty before using charAt(0)
2.	Error: NullPointerException	Rectified it by ensuring the string is not null; use isEmpty() before accessing characters

Important points:

1)Always trim user inputs (.trim()) to remove leading/trailing spaces which may cause unexpected behavior.

2)Ensure names are not left empty, and age is a valid positive integer.

3)You might want to use .toUpperCase() or .toLowerCase() for consistency in the generated password.

4)For better structure, use a separate Student class with attributes and a generatePassword() method.

Program 2:

Aim: Design and implement a java program that will do the following operations to this string “Welcome ! you are practicing strings concept”

i)Convert all 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

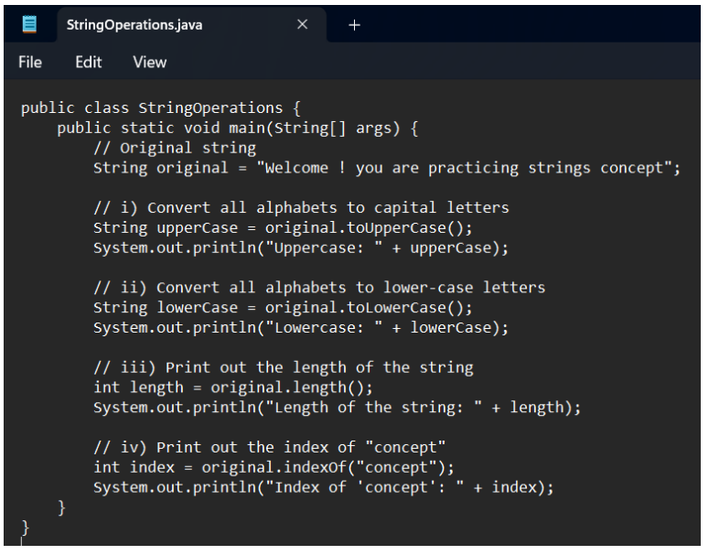
iv)Print out the index of concept

Class diagram:

StringOperations
- original: String

Step 1: Open notepad<<save the note pad in the path[desktop<<oops<<week 10<< StringOperations.java.

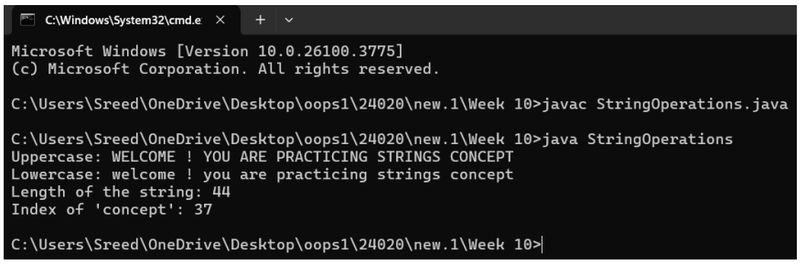
Code:



Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac StringOperations.java <<java StringOperations the program is executed successfully.

Output:



Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: StringIndexOutOfBoundsException	Rectified it by checking the index using string.length() before accessing characters
2.	Error: main method not found in class	Rectified it by using public static void main(String[] args)

Important points:

1)Strings are immutable

2)String index starts with 0 not with 1.

3)Length includes spaces and special characters.

Program 3:

Aim: Implement a java program using the below array method

i)Sorting the elements(numbers & strings)of array

ii)Convert the array elements into strings

iii)Fill the part of any array

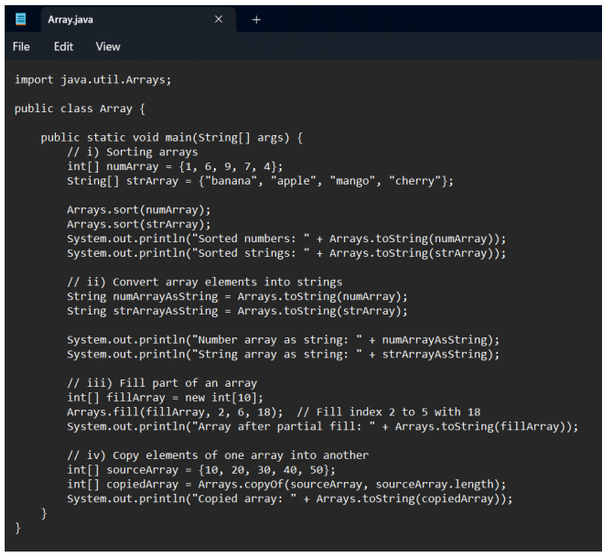
iv)Copy the elements of one array into another

Class diagram:

Array
+main(String [] args):void
+stringArray:String[] +numberArray:int[] +fillArray:int[] +sourceArray,destArray:int[]

Step 1: Open notepad<<save the note pad in the path[desktop<<oops<<week 10<<Array.java.

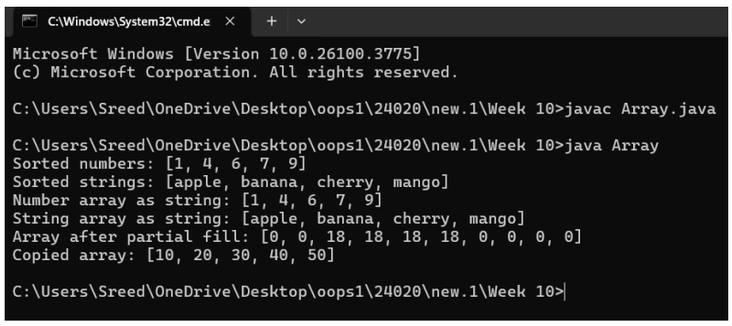
Code:



Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac Array.java <<java Array the program is executed successfully.

Output:



Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: ArrayIndexOutOfBoundsException	Rectified it by checking the array.length before accessing
2.	Error: NullPointerException	Ensuring array is properly initialized before use

Important points:

Sorts arrays of numbers or strings in ascending order. For custom order, use Comparator.

First element is at index 0. Last is at array.length - 1.

Uninitialized int[] elements are 0; for objects like String[], it's null.

Arrays.toString() : Converts array contents to a readable string format like [1, 2, 3].

Program 4:

Aim: Implement a java program using the below array list methods

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

v)Clear the elements from the array list

Class Diagram:

four
+list:ArrayList<String>
+main(String[] args):void

Step 1: Open notepad<<save the note pad in the path[desktop<<oops<<week 10<<four.java.

Code:

four.java

File Edit View

```
import java.util.ArrayList;
public class four{
        public static void main(String[] args){
        ArrayList<String> bikes = new ArrayList<>();
        bikes.add("Honda");
        bikes.add("Royal_Enfield");
        bikes.add("Truimph");
         System.out.println("Original List: " + bikes);
        // i) Insert an element at a particular index
        bikes.add(1, "Yamaha");
         System.out.println("After inserting : " + bikes);

        // ii) Modify an element in the ArrayList
        bikes.set(2,"KTM");
         System.out.println("After modifying : " + bikes);

         // iii) Access an element from the ArrayList
        String bikesAtIndex3 = bikes.get(3);
         System.out.println("Element at index 3: " + bikesAtIndex3);

         // iv) Remove an element from the ArrayList
        bikes.remove("Honda");
        System.out.println("After removing :" + bikes);

         // v) Clear all elements from the ArrayList
        bikes.clear();
        System.out.println("After clearing the list: " + bikes);
    }
}
```

Step 2: Copy the path then paste it in command prompt and run it.

Step 3: Enter the commands as javac four.java <<java four the program is executed successfully.

Output:

C:\Windows\System32\cmd.e

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

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week 10>javac four.java

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week 10>java four
Original List: [Honda, Royal_Enfield, Truimph]
After inserting : [Honda, Yamaha, Royal_Enfield, Truimph]
After modifying : [Honda, Yamaha, KTM, Truimph]
Element at index 3: Truimph
After removing :[Yamaha, KTM, Truimph]
After clearing the list: []

C:\Users\Sreed\OneDrive\Desktop\oops1\24020\new.1\Week 10>
```

Step 4: After running the program system automatically creates a class of it.

Error:

S.NO	ERROR MESSAGE	ERROR RECTIFICATION
1.	Error: IndexOutOfBoundsException	Rectified it by changing the list.size() before accessing.
2.	Error: Forgetting import statement	Rectified it by adding import java.util.ArrayList; at the top of the code.

Important points:

ArrayList automatically resizes itself when elements are added or removed.

Use add(index, element) to insert at a specific position. Elements are shifted.

Use set(index, element) to update an element at a specific index.

Use get(index) to retrieve an element.

You can declare ArrayList<String>, ArrayList<Integer>, etc. to enforce type safety.