

Intro to Java (CSL215) Lab Workbook



Faculty name Dr.Swati Gupta

Student name:Himanshu Saini

Roll No: 24csu350

Semester:3rd Sem

Group: 08

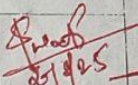
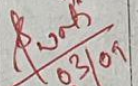
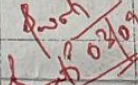
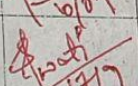
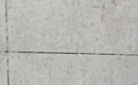
Department of Computer Science and Engineering

The NorthCap University, Gurugram- 122017, India

Session 2025-26



Index

S.No	Experiment	Date of Experiment	Date of Submission	Student Sign	Marks	CO Covered	Signature
1	Prog. on DataTypes	20/08/25	20/08/25	Himanshu		CO1	
2	Prog. on Control Statements	27/08/25	27/08/25	Himanshu		CO1	
3	Prog. on Arrays	28/08/25	31/09/25	Himanshu		CO1	
4	Prog. on Classes & Objects	03/09/25	10/09/25	Himanshu		CO2	
5	Prog. on Inheritance	10/09/25	17/09/25	Himanshu		CO3	
6	Prog. on Packages						
7	Prog. on Interfaces						
8	Prog. on Exception Handling						

PRACTICAL NO. 1

Student Name and Roll Number: Himanshu Saini and 24csu350
Semester /Section: 3 rd /D
Link to Code:
Date: 20/08/2025
Faculty Signature:

Objective(s):

- Perform variable assignment.
- Use comments in code
- Use operator precedence and operator associativity.
- Effectively use arithmetic expressions in Java

Outcome:

Student will be familiarizing with the data types in Java.

Problem Statement:

Q1. Write a program that takes two integers (values to be given within the program) and displays the output of the following operators: addition, subtraction, multiplication, division and modulus.

Definition of Done:

DoD 1: Assign two numbers to two variables.

DoD2: Use separate variables to store the results.

Q2. Write a Java program to convert minutes into the number of years, months and days.

Definition Of Done :

DoD 1: Ask the user to input the minutes

DoD 2: Display years and months and days in the sequence yy-mm-dd.

Q3 Write a Java program to convert temperature from Fahrenheit to Celsius. Formula:

$$C = \frac{(F - 32) * 5}{9}$$

Definition of Done (DoD):

- DoD 1: The program should ask the user to enter temperature in Fahrenheit.
- DoD 2: The program should calculate the Celsius equivalent.
- DoD 3: The program should display the result.

Q4 Write a Java program to find the sum of digits of a 3-digit number entered by the user.

Definition of Done (DoD):

- DoD 1: The program should ask the user to enter a 3-digit number.
- DoD 2: The program should extract digits using division and modulus operators.
- DoD 3: The program should display the sum of digits.

Q5. Reverse a 3-digit Number

Write a Java program to reverse a 3-digit number entered by the user.

Definition of Done (DoD):

- DoD 1: The program should ask the user to enter a 3-digit number.
- DoD 2: The program should reverse the number using division and modulus operators.
- DoD 3: The program should display the reversed number.

Background Study: Data Types in Java

Primitive data types: The primitive data types include boolean, char, byte, short, int, long, float and double.

Non-primitive data types: The non-primitive data types include Classes, Interfaces, and Arrays.

Question Bank:

1. Why Java is considered dynamic?
2. What is Java Virtual Machine and how it is considered in context of Java's platform independent feature?
3. List two Java IDE's? List some Java keywords (unlike C, C++ keywords)?
4. Consider the following class:

```
public class IdentifyMyParts {  
    public static int x = 7;  
    public int y = 3;  
}
```

a) What are the class variables?

b) What are the instance variables?

5. Differentiate between instance variable and a class variable.

6. Explain how to create instance of a class by giving an example

7. What is singleton class? Where is it used?

8. What is a native method?

9. Difference between a public and a non-public class

Flipped practicals

1. What is the output from the following code:

```
IdentifyMyParts a = new IdentifyMyParts();
IdentifyMyParts b = new IdentifyMyParts();
a.y = 5;
b.y = 6;
a.x = 1;
b.x = 2;
System.out.println("a.y = " + a.y);
System.out.println("b.y = " + b.y);
System.out.println("a.x = " + a.x);
System.out.println("b.x = " + b.x);
System.out.println("IdentifyMyParts.x = " + IdentifyMyParts.x);
```

2. What's wrong with the following program?

```
public class SomethingIsWrong {
    public static void main(String[] args) {
        Rectangle myRect;
        myRect.width = 40;
        myRect.height = 50;
        System.out.println("myRect's area is " + myRect.area());
    }
}
```

--

Student Work Area

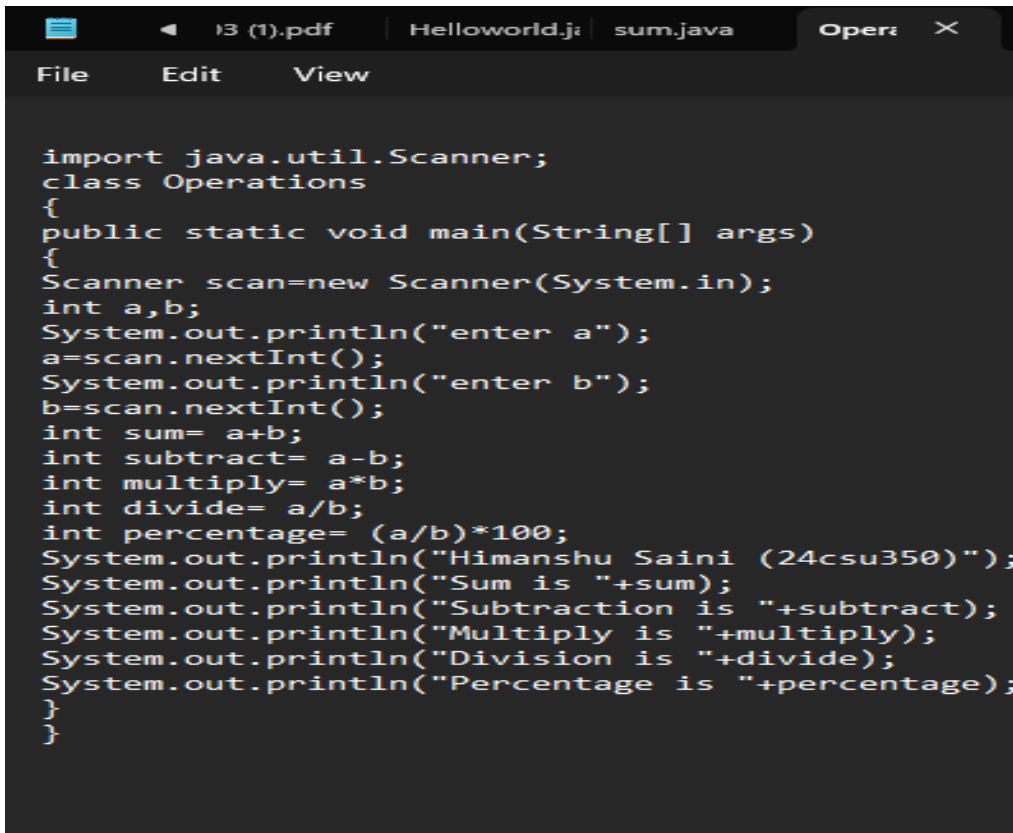
Algorithm/Flowchart/Code/Sample Outputs

```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

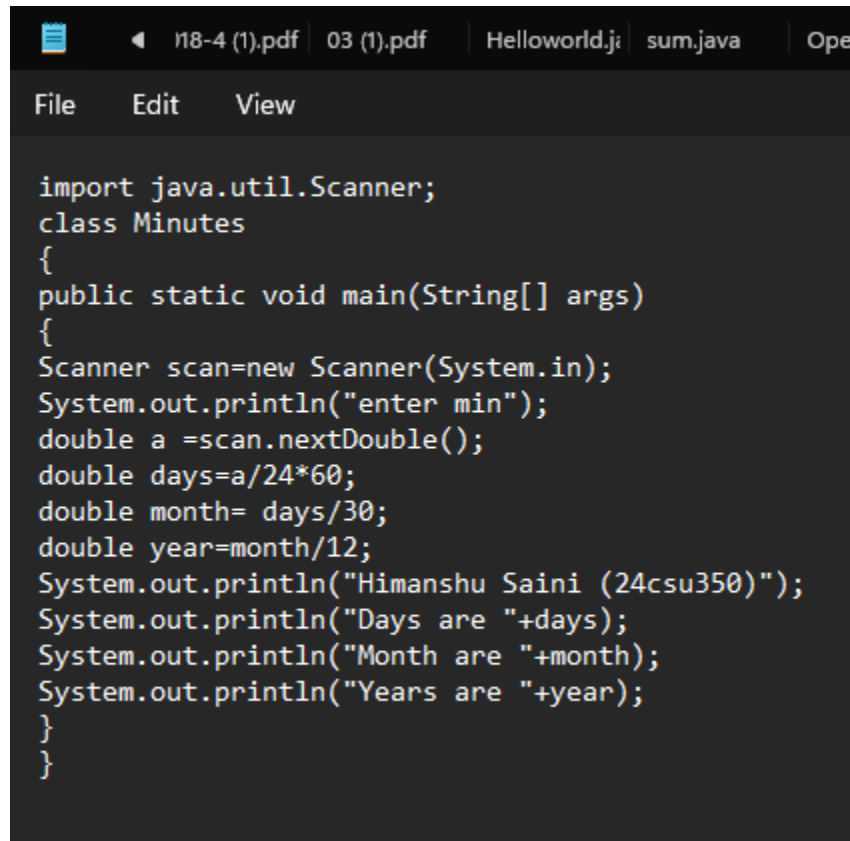
C:\Users\ace\OneDrive\Pictures\java>javac Operations.java

C:\Users\ace\OneDrive\Pictures\java>java Operations
enter a
5
enter b
6
Himanshu Saini (24csu350)
Sum is 11
Subtraction is -1
Multiply is 30
Division is 0
Percentage is 0

C:\Users\ace\OneDrive\Pictures\java>
```



```
import java.util.Scanner;
class Operations
{
    public static void main(String[] args)
    {
        Scanner scan=new Scanner(System.in);
        int a,b;
        System.out.println("enter a");
        a=scan.nextInt();
        System.out.println("enter b");
        b=scan.nextInt();
        int sum= a+b;
        int subtract= a-b;
        int multiply= a*b;
        int divide= a/b;
        int percentage= (a/b)*100;
        System.out.println("Himanshu Saini (24csu350)");
        System.out.println("Sum is "+sum);
        System.out.println("Subtraction is "+subtract);
        System.out.println("Multiply is "+multiply);
        System.out.println("Division is "+divide);
        System.out.println("Percentage is "+percentage);
    }
}
```

```
import java.util.Scanner;
class Minutes
{
    public static void main(String[] args)
    {
        Scanner scan=new Scanner(System.in);
        System.out.println("enter min");
        double a =scan.nextDouble();
        double days=a/24*60;
        double month= days/30;
        double year=month/12;
        System.out.println("Himanshu Saini (24csu350)");
        System.out.println("Days are "+days);
        System.out.println("Month are "+month);
        System.out.println("Years are "+year);
    }
}
```

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac Minutes.java

C:\Users\ace\OneDrive\Pictures\java>java Minutes
enter min
55
Himanshu Saini (24csu350)
Days are 137.5
Month are 4.583333333333333
Years are 0.3819444444444444

C:\Users\ace\OneDrive\Pictures\java>0|
```

```
118-4 (1).pdf | 03 (1).pdf | Helloworld.ji | sum.java | Operations.j | Cho
File Edit View

import java.util.Scanner;
class f_to_c
{
public static void main(String[] args)
{
Scanner scan=new Scanner(System.in);
System.out.println("enter C ");
double C =scan.nextDouble();
double f=C*9/5+32;
System.out.println("Himanshu Saini(24csu350) ");
System.out.println("fahrenheit are "+f);
}
}
```

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>java f_to_c.java
enter C
55
Himanshu Saini(24csu350)
fahrenheit are 131.0

C:\Users\ace\OneDrive\Pictures\java>
```

```
C:\Users\ace\OneDrive\Pictures\java2\Sum.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Rev.java Sum.java MaxMin.java
1  import java.util.Scanner;
2  class Sum
3  {
4  public static void main(String[] arg)
5  {
6  Scanner scan=new Scanner(System.in);
7
8  System.out.println("Enter a number");
9  int n =scan.nextInt();
10 int sum=0;
11 while(n>0)
12 { int digit = n % 10;
13  sum = sum+digit;
14  n=n/10;
15
16  }
17 System.out.println("Himanshu Saini(24csu350)");
18 System.out.println("sum of digit is " +sum);
19 }
20 }
```

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java2>javac Sum.java

C:\Users\ace\OneDrive\Pictures\java2>java Sum
Enter a number
5
Himanshu Saini(24csu350)
sum of digit is 5

C:\Users\ace\OneDrive\Pictures\java2>java Sum
Enter a number
564
Himanshu Saini(24csu350)
sum of digit is 15

C:\Users\ace\OneDrive\Pictures\java2>
```

```
C:\Users\ace\OneDrive\Pictures\java2\Rev.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Rev.java Sum.java MaxMin.java
C:\Users\ace\OneDrive\Pictures\java2\Rev.java
1 import java.util.Scanner;
2 class Rev
3 {
4     public static void main(String[] arg)
5     {
6         Scanner scan=new Scanner(System.in);
7         System.out.println("Enter a number");
8         int n =scan.nextInt();
9         int reverse=0;
10        while(n>0)
11        { int digit = n % 10;
12          reverse = reverse*10+digit;
13          n=n/10;
14        }
15        System.out.println("Himanshu Saini(24csu350)");
16        System.out.println("reverse number is " +reverse);
17    }
18 }
19 }
```

```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java2>javac Rev.java

C:\Users\ace\OneDrive\Pictures\java2>java Rev
Enter a number
654321
Himanshu Saini(24csu350)
reverse number is 123456

C:\Users\ace\OneDrive\Pictures\java2>
```

Ans1. Java is considered dynamic because it can adapt while the program is running: it can load classes at runtime, automatically manage memory with garbage collection, and decide which method to execute through polymorphism and late binding. This means Java doesn't need everything fixed at compile time; instead, it adjusts and links things during execution, making programs more flexible and powerful.

Ans2. The Java Virtual Machine (JVM) is like a special computer inside your real computer that runs Java programs. When you write Java code, it is first converted into bytecode (a kind of universal language). This bytecode is not specific to any operating system like Windows, Linux, or Mac. Instead, the JVM on each system reads and runs this bytecode. Because every operating system has its own JVM, the same Java program can run anywhere without changes.

Ans3. Two popular Java IDEs (Integrated Development Environments):

Eclipse

IntelliJ IDEA (also NetBeans is common)

Some Java keywords (these are special reserved words in Java, not used as names):

Class

Interface

extends

implements

import

package

throws

instanceof

Ans4. x (static) → only one copy for the whole class. If you change it in one object, it changes for all.

y (non-static) → each object has its own copy. Changing it in one object does not affect the other.

static = shared by all objects.

normal variable = separate for each object.

Ans5. Instance Variable

Declared without static.

Each object gets its own separate copy.

Changing it in one object does not affect other objects.

Example: int age; → each student object can have a different age.

Class Variable (Static Variable)

Declared with static.

Only one copy exists for the whole class.

Shared by all objects.

Example: static String schoolName; → all students share the same school name.

In short:

Instance variable = personal property of each object.

Class variable = common property shared by all objects.

Ans6. 6.In java ,an instance of a class is simply an object.

It can be created using the “new” keyword,which calls the class constructor.

ClassName objectName = new ClassName();

7.A singleton class in java is a class that allows only one instance to be created throught the entire program. It ensures that a single ,global point of access exists for that object .

It is used for logging ,configuration setting ,database connections .

8.A native method in Java is a method that is implemented in another programming language ,not in java itself.Such methods are marked using the native keyword.They are mainly used to interact with system -level resources ,improve performance or use legacy libraries.

9.Public class can be accessed from anywhere ,a source file can have only one public class and its name must match the filename ,

Java public class

Student { }

Non Public class can be accessed only within the same package ,a source file can have multiple non public classes and their name do not need to match the filename

java class Teacher { }

Flipped practicals

Q1.Output

a.y = 5

b.y = 6

a.x = 2

b.x = 2

IdentifyMyParts.x = 2

Q2. Here, myRect is just a name (reference), but no actual Rectangle object was made.

Rectangle myRect = new Rectangle();

PRACTICAL NO. 2

Student Name and Roll Number: Himanshu Saini and 24csu350
Semester /Section: 3rd/D
Link to Code:
Date:27/08/2025
Faculty Signature:

Objective

To familiarize the students with control statements in JAVA

Program Outcome

Through this practical, students will learn about the control statements

Problem Statement

1. Write a Java program to display the character rhombus structure. You can assume a character to create rhombus/diamond, left and right triangle, pyramid.

Definition of Done:

DoD1: The program asks the user for the number of rows for the rhombus structure from the user

DoD 2: After a successful run, the program should ask the user if he wants to try the program for some other value, and should exit only when the user wants to exit (by pressing the character 'N').

2. Write a Java program to solve quadratic equations (use if, else if and else).

Definition of Done:

DoD 1: The program asks the values of coefficients of a quadratic equation.

DoD 2: The program should display the roots or an appropriate message.

3. Write a Java program that asks the user to provide a single character from the alphabet. Print Vowel or Consonant, depending on the user input. If the user input is not a letter (between a and z or A and Z), or is a string of length > 1, print an error message.

Definition of Done:

DoD 1: The program asks an input from the user.

DoD 2: A single character input is taken from the user or an error message is generated.

DoD 3: The program should print "Vowel" if the entered character is a vowel and "Consonant" if the entered character is a consonant.

4. Write a Java program to check whether a number entered by the user is **prime or not** using control statements.

Definition of Done (DoD):

- DoD 1: The program should ask the user to enter an integer.
 - DoD 2: The program should check whether the number is prime using loops/conditionals.
 - DoD 3: The program should display a clear message: "*Number is Prime*" or "*Number is Not Prime*".
5. Write a Java program to generate and display the **Fibonacci series** up to n terms using control statements:

Definition of Done (DoD):

- DoD 1: The program should ask the user to enter the number of terms (n).
- DoD 2: The program should use only loops/conditionals to generate the series (no functions).
- DoD 3: The program should display the series in proper format.

Background Study:

Java compiler executes the java code from top to bottom. The statements are executed according to the order in which they appear. However, Java provides statements that can be used to control the flow of java code. Such statements are called control flow statements.

Java provides three types of control flow statements.

- Decision Making statements
- Loop statements
- Jump statements

Question Bank

1. The most basic control flow statement supported by the Java programming language is the ____ statement.
2. The ____ statement allows for any number of possible execution paths.
3. The ____ statement is similar to the while statement, but evaluates its expression at the ____ of the loop.
4. How do you write an infinite loop using the for statement?
5. How do you write an infinite loop using the while statement?
6. Which looping process checks the test condition at the end of the loop?
7. Why do we use continue statement?
8. What is the size of boolean variable?
9. Which looping process is best used when the number of iterations is known?

Flipped Practicals

1. Consider the following code snippet.

```
if (aNumber >= 0)
```

```
if (aNumber == 0)
```

```
System.out.println("first string");
```

```
else System.out.println("second string");
```

```
System.out.println("third string");
```

- a) What output do you think the code will produce if aNumber is 3?
- b) Write a test program containing the previous code snippet; make aNumber 3. What is the output of the program? Is it what you predicted? Explain why the output is what it is; in other words, what is the control flow for the code snippet?
- c) Using only spaces and line breaks, reformat the code snippet to make the control flow easier to understand.
- d) Use braces, { and }, to further clarify the code.

2. What's wrong? `for (int k = 2, k <= 12, k++)`

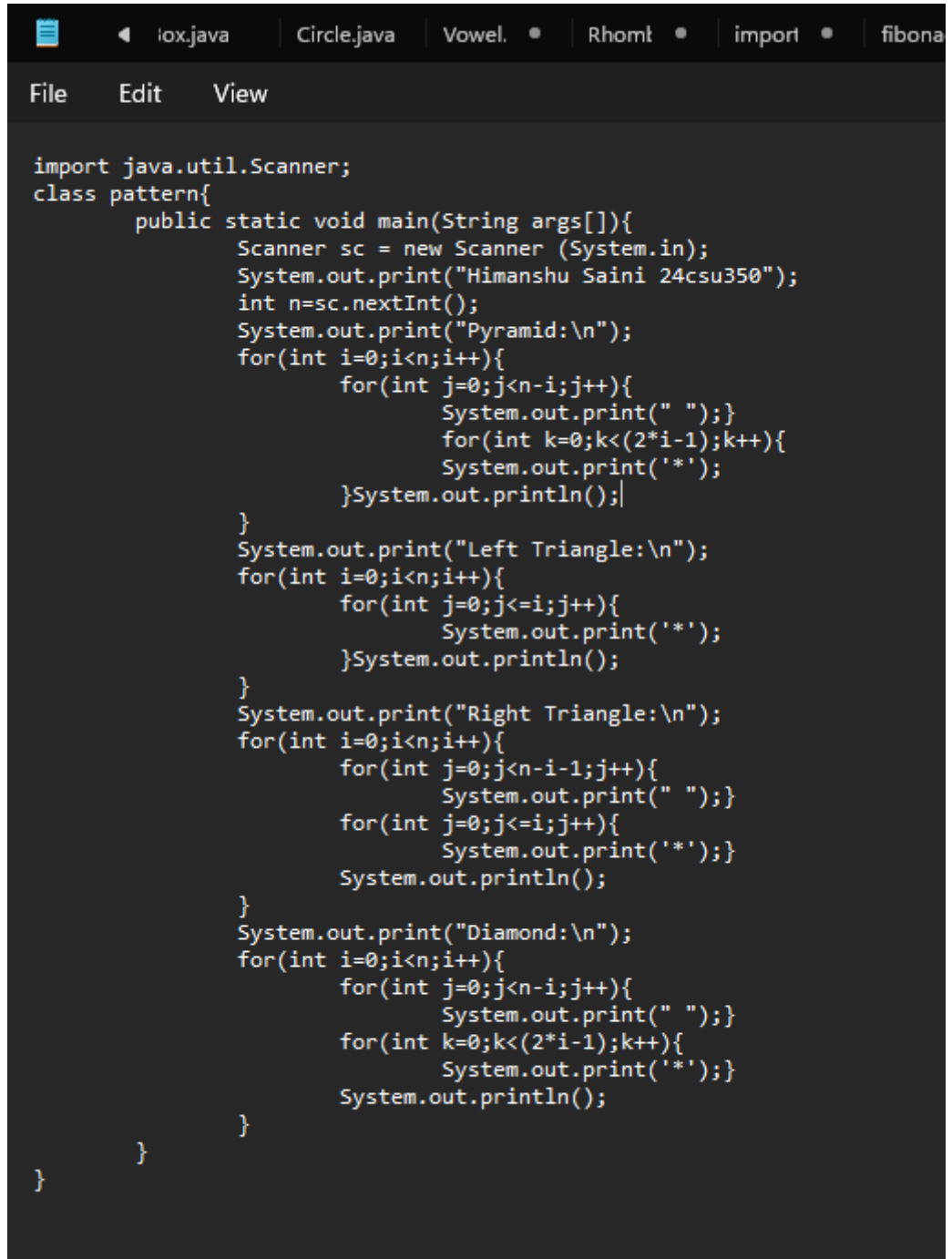
3. If there is more than one statement in the block of a for loop, what must be placed at the beginning and the ending of the loop block?

4. What value is stored in num at the end of this looping?

```
for (num = 1; num <= 5; num++)
```

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs



```
lox.java | Circle.java | Vowel. | Rhomt. | import. | fibona
File Edit View

import java.util.Scanner;
class pattern{
    public static void main(String args[]){
        Scanner sc = new Scanner (System.in);
        System.out.print("Himanshu Saini 24csu350");
        int n=sc.nextInt();
        System.out.print("Pyramid:\n");
        for(int i=0;i<n;i++){
            for(int j=0;j<n-i;j++){
                System.out.print(" ");
                for(int k=0;k<(2*i-1);k++){
                    System.out.print('*');
                }System.out.println();
            }
            System.out.print("Left Triangle:\n");
            for(int i=0;i<n;i++){
                for(int j=0;j<=i;j++){
                    System.out.print('*');
                }System.out.println();
            }
            System.out.print("Right Triangle:\n");
            for(int i=0;i<n;i++){
                for(int j=0;j<n-i-1;j++){
                    System.out.print(" ");
                }
                for(int j=0;j<=i;j++){
                    System.out.print('*');
                }
                System.out.println();
            }
            System.out.print("Diamond:\n");
            for(int i=0;i<n;i++){
                for(int j=0;j<n-i;j++){
                    System.out.print(" ");
                }
                for(int k=0;k<(2*i-1);k++){
                    System.out.print('*');
                }
                System.out.println();
            }
        }
    }
}
```

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac pattern.

C:\Users\ace\OneDrive\Pictures\java>java pattern
Himanshu Saini 24csu350

5
Pyramid:

    *
   ***
  *****
 *****
Left Triangle:
*
**
***
****
*****
Right Triangle:
    *
   **
  ***
 ****
 *****
Diamond:

    *
   ***
  *****
 *****
```

```
pattern.java | isprime.java | sumele.java | revarr | SearchElem | import | MaxMin

File Edit View

public class QuadraticEquation {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Himanshu Saini 24csu350");

        System.out.print("Enter coefficient a: ");
        double a = sc.nextDouble();

        System.out.print("Enter coefficient b: ");
        double b = sc.nextDouble();

        System.out.print("Enter coefficient c: ");
        double c = sc.nextDouble();
        double discriminant = b * b - 4 * a * c;

        System.out.println("\nEquation: " + a + "x^2 + " + b + "x + " + c + " = 0");

        if (discriminant > 0) {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("Two distinct real roots:");
            System.out.println("Root 1 = " + root1);
            System.out.println("Root 2 = " + root2);
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("One real root (repeated): " + root);
        } else {
            System.out.println("No real roots. Discriminant < 0, roots are imaginary");
        }

        sc.close();
    }
}
```

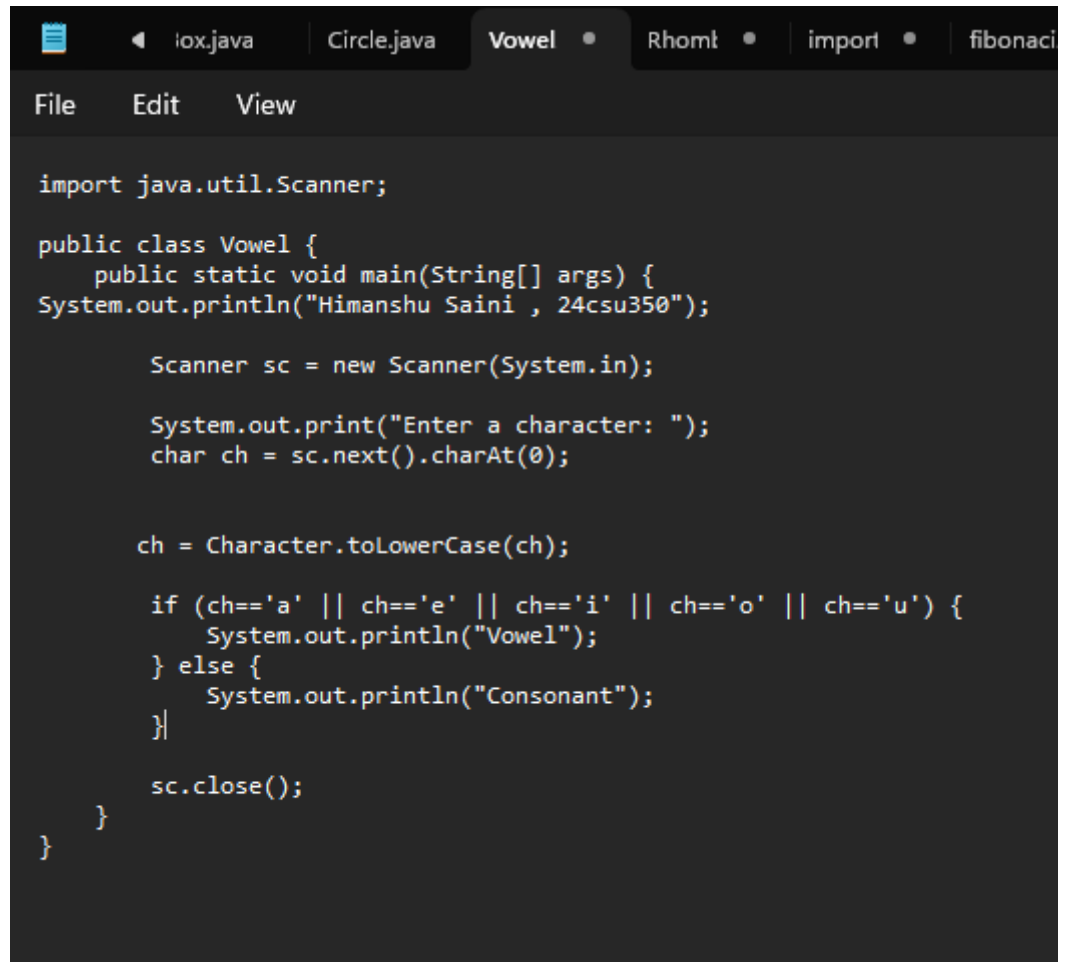
```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac QuadraticEquation.java

C:\Users\ace\OneDrive\Pictures\java>java QuadraticEquation
Himanshu Saini 24csu350Enter coefficient a: 5
Enter coefficient b: 4
Enter coefficient c: 8

Equation: 5.0x^2 + 4.0x + 8.0 = 0
No real roots. Discriminant < 0, roots are imaginary.

C:\Users\ace\OneDrive\Pictures\java>
```



```
import java.util.Scanner;

public class Vowel {
    public static void main(String[] args) {
        System.out.println("Himanshu Saini , 24csu350");

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a character: ");
        char ch = sc.next().charAt(0);

        ch = Character.toLowerCase(ch);

        if (ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u') {
            System.out.println("Vowel");
        } else {
            System.out.println("Consonant");
        }

        sc.close();
    }
}
```

```
C:\Windows\System32\cmd.e  X  +  v

Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac Vowel.java

C:\Users\ace\OneDrive\Pictures\java>java Vowel
Himanshu Saini , 24csu350
Enter a character: Lavisha
Himanshu Saini , 24csu350
Consonant

C:\Users\ace\OneDrive\Pictures\java>
```

```
1 import java.util.Scanner;
2 class isprime{
3     public static void main(String args[]){
4         Scanner sc = new Scanner (System.in);
5         System.out.print("Himanshu Saini 24csu350");
6         System.out.print("Enter a no. = ");
7         int a = sc.nextInt();
8         int count = 0;
9         for(int i=2;i<=(a-1);i++){
10             if(a%i==0){
11                 count++;
12             }else{
13                 continue;
14             }
15         }
16         if(count==0){
17             System.out.print("Entered number is a prime number");
18         }else{
19             System.out.print("Entered number is not a prime number");
20         }
21     }
22 }
```

```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac isprime.java

C:\Users\ace\OneDrive\Pictures\java>java isprime
Himanshu Saini 24csu350Enter a no. = 54
Entered number is not a prime number
C:\Users\ace\OneDrive\Pictures\java>
```

```

J isprime.java  J fibonacci.java X
C: > Users > ace > OneDrive > Pictures > java > J fibonacci.java
1  import java.util.Scanner;
2  class fibonacci
3  {
4      public static void main(String args[]){
5          Scanner sc = new Scanner (System.in);
6          System.out.print("Himanshu Saini 24csu350 ");
7              int a=0;
8              int b=1;
9              int c;
10             System.out.print("Enter no. upto which fibonacci series have to print = ");
11             int n=sc.nextInt();
12             System.out.print(a + b + " ");
13             for(int i=0;i<n;i++){
14                 c = a+b;
15                 System.out.print(c + " ");
16                 a=b;
17                 b=c;
18             }
19         }
20     }

```

```

C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac fibonacci.java

C:\Users\ace\OneDrive\Pictures\java>java fibonacci
Himanshu Saini 24csu350 Enter no. upto which fibonacci series have to print = 5
1 1 2 3 5 8
C:\Users\ace\OneDrive\Pictures\java>

```

Question Bank

A1. if statement

A2. switch statement

A3. do-while statement

A4. (;) { }

A5. while(true) { }

A6. do-while loop

A7. To skip the current iteration and move to the next loop iteration

A8. 1 byte

A9. For loop

Flipped Practicals

A1. a) second string

third string

b) second string

third string

c) if (aNumber >= 0)

if (aNumber == 0)

System.out.println("first string");

else

System.out.println("second string");

System.out.println("third string");

d) if (aNumber >= 0) {

if (aNumber == 0) {

System.out.println("first string");

} else {

System.out.println("second string");

}

}

System.out.println("third string");

A2. Commas are used instead of semicolans

A3. Curly braces around the block {}

A4. 6

PRACTICAL NO. 3**Student Name and Roll Number: Himanshu Saini and 24csu350****Semester /Section:3rd/D****Link to Code:****Date: 03/09/2025****Faculty Signature:****Objective**

To familiarize the students with array in JAVA.

Program Outcome

The students will learn the concept of arrays in Java.

Problem Statement

1. Write a Java program to find the maximum and minimum value of an array.

Definition of Done:

DoD 1: The program should ask the user to enter the elements of the array.

DoD 2: The program should display the maximum and minimum elements of the array.

2. Write a Java program to find the index of an array element in an array of size n. The program should not use any function other than main () functions.

Definition of Done:

DoD 1: The program should ask the user to enter the elements of the array.

DoD 2: The program should ask the user to enter a number to search.

DoD 3: The program should display the elements of the array entered.

DoD 4: The program should display the index of the number if the item is present or display -1 if the element is not present.

3. Write a Java Program to count even and odd numbers in an array.

Definition of Done

DoD 1: The program should ask the user to enter the elements of the array.

DoD 2: Even elements will be stored in EvenArray[] and odd elements will be stored in oddArray[].

DoD 3: Display all three arrays along with their length.

4. Write a Java program to read numbers in an integer array of size 5 and display the following (using functions for each functionality):

i) Sum of all the elements

ii) Sum of alternate elements in the array.

Definition of Done

DoD 1: The program should ask the user to enter the elements of the array.

DoD 2: The program should display a menu with the above choices and ask the user to choose one of the choices.

5. Write a Java program to read numbers in an integer array of size 5 and display the array in **reverse order**:

Definition of Done (DoD):

DoD 1: The program should ask the user to enter the elements of the array.

DoD 2: The program should display the reversed array.

6. Write a Java program to read numbers in an integer array of size 5 and **remove duplicate elements** from the array.

Definition of Done (DoD):

DoD 1: The program should ask the user to enter the elements of the array.

DoD 2: The program should display the array after removing duplicates.

Background Study:

Java array is an object which contains elements of a similar data type. Additionally, The elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.

Array in Java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and so on.

Question Bank:

1. Can you pass the negative number as an array size?
2. Can you change the size of the array once you define it?
3. What is an anonymous array?
4. What is the difference between `int[] a` and `int a[]` ?
5. What are jagged arrays in java? Give example.

Flipped Practicals

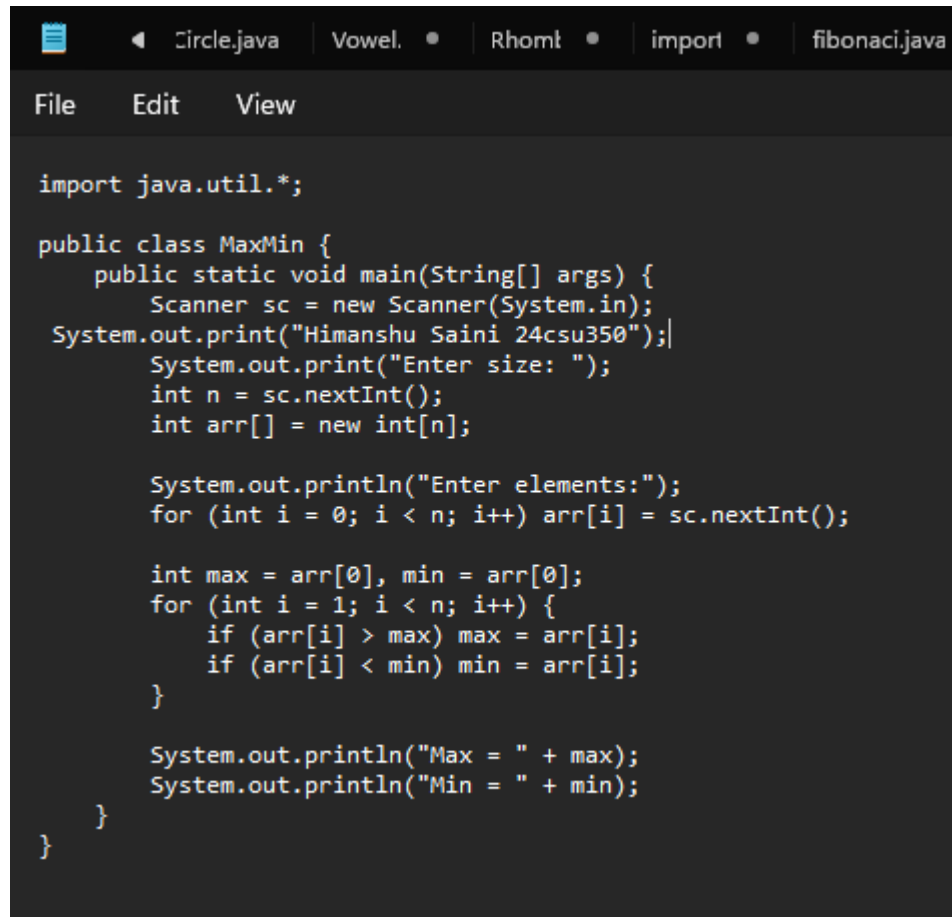
1. Which of these is an incorrect array declaration?
a) `int arr[] = new int[5]`
b) `int [] arr = new int[5]`
c) `int arr[] = new int[5]`
d) `int arr[] = int [5] new`
2. What will be the output of the following program?

```
public class MyFirst {  
    public static void main(String[] args) {  
        MyFirst obj = new MyFirst(n);  
    }  
    static int a = 10;  
    static int n;  
    int b = 5;  
    int c;  
    public MyFirst(int m) {  
        System.out.println(a + ", " + b + ", " + c + ", " + n + ", " + m);  
    }  
}
```

```
}  
// Instance Block  
{  
    b = 30;  
    n = 20;  
}  
// Static Block  
static  
{  
    a = 60;  
}  
}
```

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs



```
import java.util.*;

public class MaxMin {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Himanshu Saini 24csu350");
        System.out.print("Enter size: ");
        int n = sc.nextInt();
        int arr[] = new int[n];

        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++) arr[i] = sc.nextInt();

        int max = arr[0], min = arr[0];
        for (int i = 1; i < n; i++) {
            if (arr[i] > max) max = arr[i];
            if (arr[i] < min) min = arr[i];
        }

        System.out.println("Max = " + max);
        System.out.println("Min = " + min);
    }
}
```

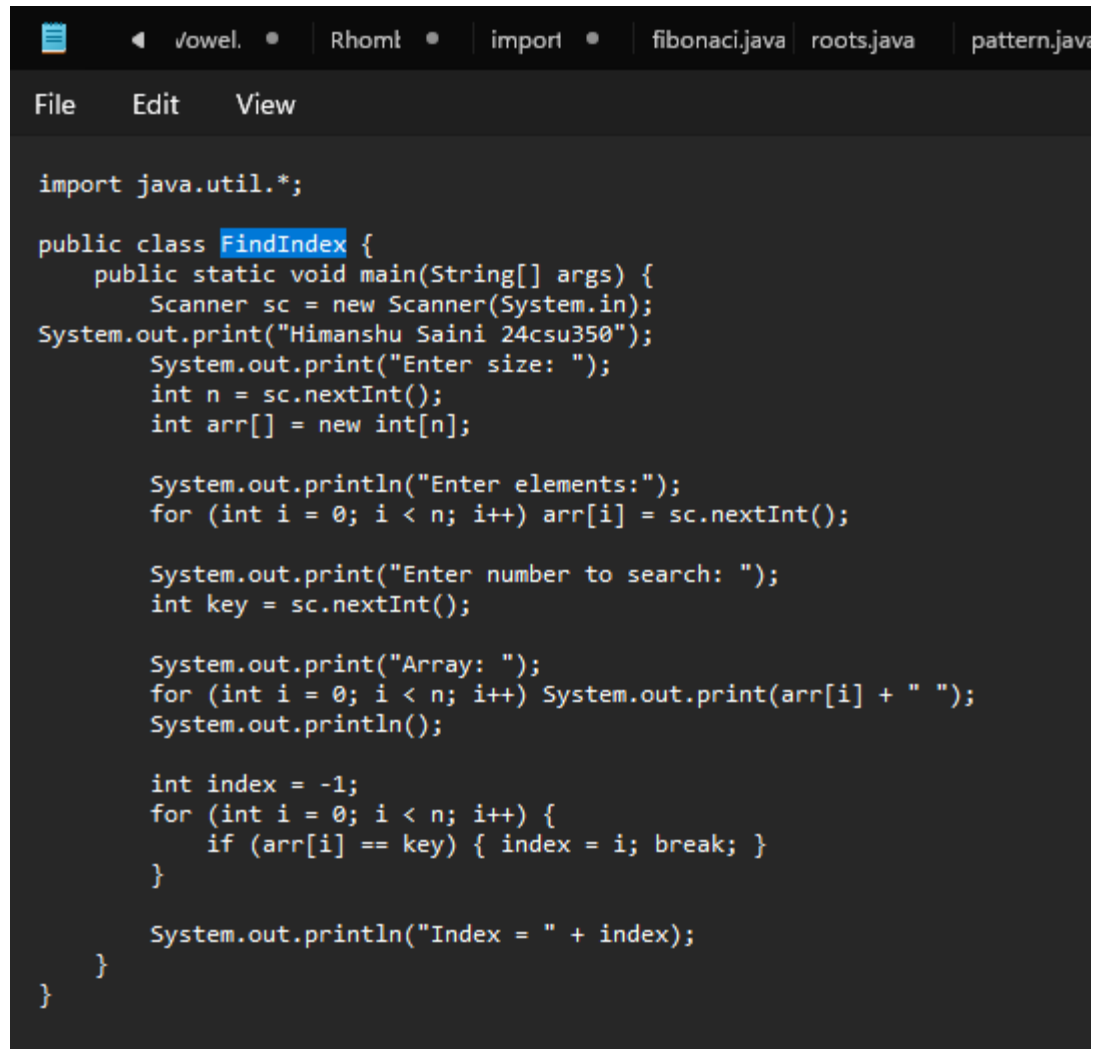
```
C:\Windows\System32\cmd.e  X  +  v

Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac MaxMin.java

C:\Users\ace\OneDrive\Pictures\java>java MaxMin
Himanshu Saini 24csu350Enter size: 2
Enter elements:
56
58
Max = 58
Min = 56

C:\Users\ace\OneDrive\Pictures\java>
```

```
import java.util.*;

public class FindIndex {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Himanshu Saini 24csu350");
        System.out.print("Enter size: ");
        int n = sc.nextInt();
        int arr[] = new int[n];

        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++) arr[i] = sc.nextInt();

        System.out.print("Enter number to search: ");
        int key = sc.nextInt();

        System.out.print("Array: ");
        for (int i = 0; i < n; i++) System.out.print(arr[i] + " ");
        System.out.println();

        int index = -1;
        for (int i = 0; i < n; i++) {
            if (arr[i] == key) { index = i; break; }
        }

        System.out.println("Index = " + index);
    }
}
```

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac FindIndex.java

C:\Users\ace\OneDrive\Pictures\java>java FindIndex
Himanshu Saini 24csu350Enter size: 3
Enter elements:
1254
125
564
Enter number to search: 2
Array: 1254 125 564
Index = -1

C:\Users\ace\OneDrive\Pictures\java>
```

```
import java.util.Scanner;
class oddeven{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.print("Himanshu Saini 24csu350");
        System.out.print("Enter the size of array = ");
        int n = sc.nextInt();
        int arr[] = new int[n];
        System.out.print("Enter the elements = ");
        for(int i=0;i<n;i++){
            arr[i]=sc.nextInt();
        }
        System.out.print("Original Array = ");
        for(int i=0;i<n;i++){
            System.out.print(arr[i]+" ");
        }
        int arr2[] = new int[n];
        int arr3[] = new int[n];
        int even = 0, odd = 0, evencount = 0, oddcount = 0;
        for(int i=0; i<n; i++){
            if(arr[i] % 2 == 0){
                arr2[evencount++] = arr[i];
                even++;
            }else {
                arr3[oddcount++] = arr[i];
                odd++;
            }
        }
        System.out.print("\nEven elements = ");
        for(int i=0; i<evencount; i++){
            System.out.print(arr2[i] + " ");
        }
        System.out.print("\nOdd elements = ");
        for(int i=0; i<oddcount; i++){
            System.out.print(arr3[i] + " ");
        }
        System.out.print("\nTotal Even numbers = " + even);
        System.out.print("\nTotal Odd numbers = " + odd);
    }
}
```

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac oddeven.java

C:\Users\ace\OneDrive\Pictures\java>java oddeven
Himanshu Saini 24csu350Enter the size of array = 3
Enter the elements = 12
45
35
Original Array = 12 45 35
Even elements = 12
Odd elements = 45 35
Total Even numbers = 1
Total Odd numbers = 2
C:\Users\ace\OneDrive\Pictures\java>
```

```
import java.util.*;

public class VeryEasyArraySum {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Himanshu Saini 24csu350");
        int arr[] = new int[5];
        int sumAll = 0, sumAlt = 0;

        System.out.println("Enter 5 numbers:");
        for (int i = 0; i < 5; i++) {
            arr[i] = sc.nextInt();
            sumAll += arr[i];
            if (i % 2 == 0) sumAlt += arr[i];
        }

        System.out.println("Sum of all = " + sumAll);
        System.out.println("Sum of alternate = " + sumAlt);
    }
}
```

```
C:\Windows\System32\cmd.e  X  +  v

Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>java VeryEasyArraySum.java
Himanshu Saini 24csu350Enter 5 numbers:
23
45
54
78
54
Sum of all = 254
Sum of alternate = 131

C:\Users\ace\OneDrive\Pictures\java>
```

```
sumele.java | revarr | SearchElem | import |
File Edit View

import java.util.Scanner;

public class ReverseArray {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Himanshu Saini 24csu350");
        int[] arr = new int[5];

        System.out.println("Enter 5 integers:");
        for (int i = 0; i < arr.length; i++) {
            arr[i] = sc.nextInt();
        }

        System.out.print("Reversed Array: ");
        for (int i = arr.length - 1; i >= 0; i--) {
            System.out.print(arr[i] + " ");
        }

        sc.close();
    }
}
```

```
C:\Windows\System32\cmd.e  X  +  v

Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac ReverseArray.java

C:\Users\ace\OneDrive\Pictures\java>java ReverseArray
Himanshu Saini 24csu350Enter 5 numbers:
45
15
24
78
59
Reversed array:
59 78 24 15 45
C:\Users\ace\OneDrive\Pictures\java>
```

```
prime.java | sumele.java | revarra | SearchElem | import
File Edit View
public class RemoveDuplicates {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Himanshu Saini 24csu350");

        int[] arr = new int[5];

        System.out.println("Enter 5 integers:");
        for (int i = 0; i < arr.length; i++) {
            arr[i] = sc.nextInt();
        }

        int n = arr.length;
        int[] temp = new int[n];
        int newLength = 0;

        for (int i = 0; i < n; i++) {
            boolean isDuplicate = false;
            for (int j = 0; j < newLength; j++) {
                if (arr[i] == temp[j]) {
                    isDuplicate = true;
                    break;
                }
            }
            if (!isDuplicate) {
                temp[newLength] = arr[i];
                newLength++;
            }
        }

        System.out.print("Array after removing duplicates: ");
        for (int i = 0; i < newLength; i++) {
            System.out.print(temp[i] + " ");
        }

        sc.close();
    }
}
```



```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac RemoveDuplicates.java

C:\Users\ace\OneDrive\Pictures\java>java RemoveDuplicates
Himanshu Saini 24csu350Enter 5 numbers:
3
56
42
10
12
Array after removing duplicates: [3, 56, 42, 10, 12]

C:\Users\ace\OneDrive\Pictures\java>
```

Question bank

1.No, you cannot pass a negative number as an array size.

So Java allows compilation but fails at runtime with
NegativeArraySizeException

2. In Java, once you define an array, its size cannot be changed.

3.An anonymous array is an array without a name.

You create it and use it immediately (for one-time use).It is
not stored in a variable, so you can't access it later.

4. No difference between `int[] a` and `int a[]`.

`int[] a` is preferred in Java for readability and avoiding
confusion.

5. A jagged array (also called array of arrays) is a 2D array with
rows of different lengths.

Unlike a normal 2D array (matrix) where all rows have the
same number of columns,

in a jagged array, each row can have a different number of
columns.

Question bank

A1. No

A2. No

A3. An anonymous array is an array created without assigning it to a Variable

A4. No difference

A5. Jagged arrays are arrays of arrays where inner arrays can have different lengths

Flipped Practical

A1. `int arr[] = int [5] new`

A2. 60, 30, 0, 20, 0

PRACTICAL NO: 4

Student Name and Roll Number: Himanshu Saini and 24csu350
Semester /Section: 3rd/D
Link to Code:
Date: 10/09/2025
Faculty Signature:

Objective To familiarize the students with classes and objects.
Program Outcome The students will learn the concept of classes and objects.
Problem Statement 1.Create a class named 'Student' with String variable 'name' and integer variable 'roll_no'. Assign the value of roll_no as '2' and that of name as "John" by creating an object of the class Student. 2. Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' without any parameter in its constructor. 3. Write a program to print the area of a rectangle by creating a class named 'Area' taking the values of its length and breadth as parameters of its constructor and having a method named 'returnArea' which returns the area of the rectangle. Length and breadth of the rectangle are entered through the keyboard.

4. Print the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate methods for each operation whose real and imaginary parts are entered by the user.

5. Write a program to calculate the distance between two points (x1, y1) and (x2, y2). All numbers and return values should be of type double.

Definition of Done:

DoD 1: Two java files to be defined. One for class definitions and another for the application

DoD 2: A class point is defined with two float variables for x1 and x2 and the following functionality:

- i. Non-parametrized and parameterized constructors are defined.
- ii. Get and set methods are defined for all the instance variables.
- iii. Distance function is defined to calculate the distance between two points.
- iv. Display function is defined with width of 7 and precision of 2.

Write this program with a static method definition for calculating the distance between two points.

Background Study

In object-oriented programming technique, we design a program using objects and classes. An object in Java is the physical as well as a logical entity, whereas, a class in Java is a logical entity only.

An object is an instance of a class. A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.

Question Bank

1. Can we call sub class constructor from super class constructor?
2. What is constructor chaining?
3. What is No-arg constructor?
4. What happens if you keep return type for a constructor?
5. What is the use of private constructor?
6. Can we use this() and super() in a method?
7. Can we define a method with same name of class?

Flipped Practicals

1. What will be the output of the following Java program?

```
class A
```

```
{
```

```
    int i;
```

```
    int j;
```

```
    A()
```

```
{
```

```
    i = 1;
```

```
    j = 2;
```

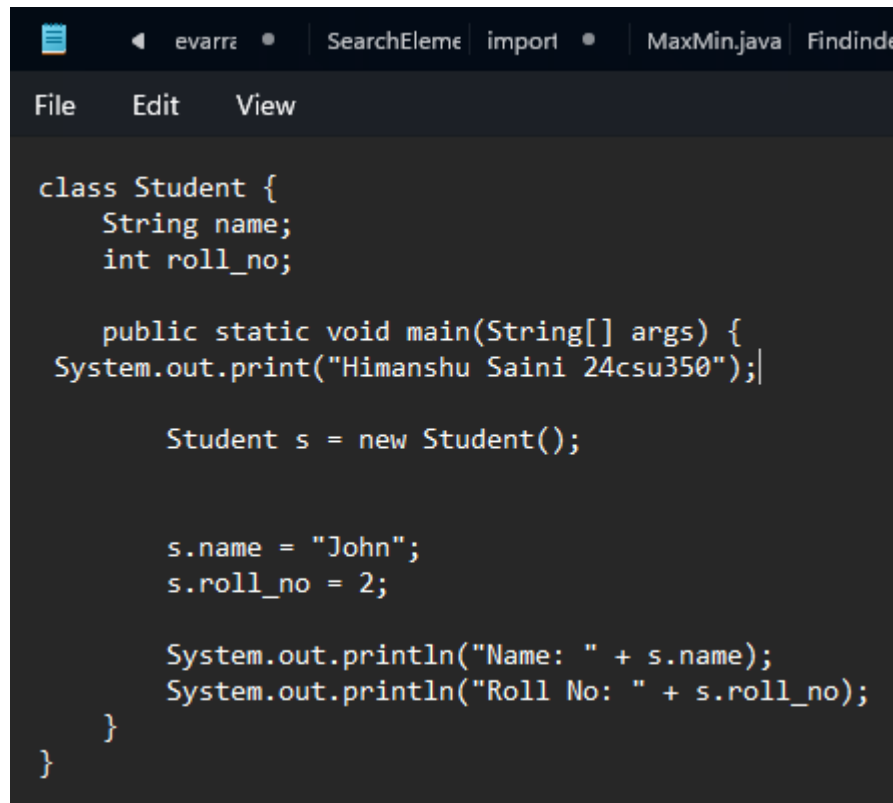
```
    }  
  
}  
  
class Output  
{  
  
    public static void main(String args[])  
  
    {  
  
        A obj1 = new A();  
  
        A obj2 = new A();  
  
        System.out.print(obj1.equals(obj2));  
  
    }  
  
}
```

- a) false
- b) true
- c) 1

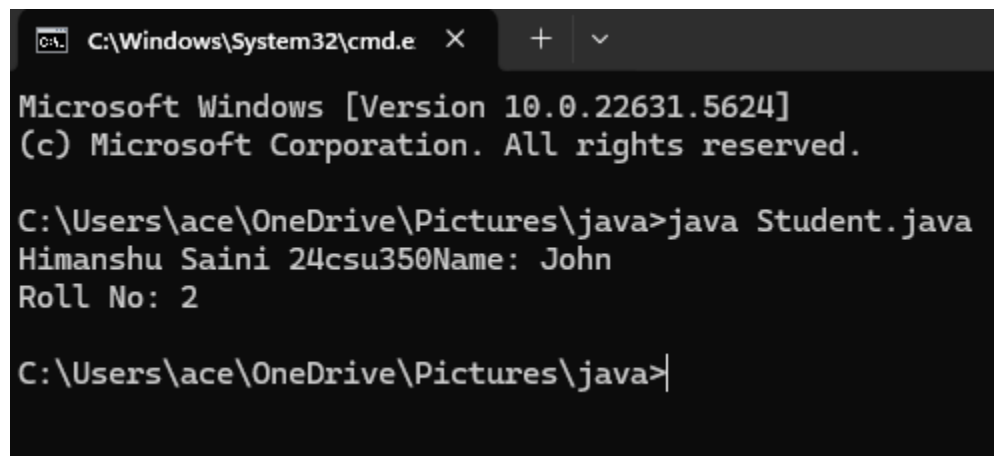
d) Compilation Error

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs



```
class Student {  
    String name;  
    int roll_no;  
  
    public static void main(String[] args) {  
        System.out.print("Himanshu Saini 24csu350");  
  
        Student s = new Student();  
  
        s.name = "John";  
        s.roll_no = 2;  
  
        System.out.println("Name: " + s.name);  
        System.out.println("Roll No: " + s.roll_no);  
    }  
}
```



```
C:\Windows\System32\cmd.e X + v  
Microsoft Windows [Version 10.0.22631.5624]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\ace\OneDrive\Pictures\java>java Student.java  
Himanshu Saini 24csu350Name: John  
Roll No: 2  
  
C:\Users\ace\OneDrive\Pictures\java>
```



```
class Triangle {
    int a, b, c;

    Triangle() {
        a = 3;
        b = 4;
        c = 5;
    }

    int perimeter() {
        return a + b + c;
    }

    double area() {
        double s = perimeter() / 2.0;
        return Math.sqrt(s * (s - a) * (s - b) * (s - c));
    }

    public static void main(String[] args) {
        System.out.print("Himanshu Saini 24csu350");
        Triangle t = new Triangle();
        System.out.println("Perimeter: " + t.perimeter());
        System.out.println("Area: " + t.area());
    }
}
```

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac Triangle.java

C:\Users\ace\OneDrive\Pictures\java>java Triangle
Himanshu Saini 24csu350Perimeter: 12
Area: 6.0

C:\Users\ace\OneDrive\Pictures\java>
```

```
import java.util.Scanner;

class Area {
    int length, breadth;

    Area(int l, int b) {
        length = l;
        breadth = b;
    }

    int returnArea() {
        return length * breadth;
    }

    public static void main(String[] args) {
        System.out.print("Himanshu Saini 24csu350");
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter length: ");
        int l = sc.nextInt();
        System.out.print("Enter breadth: ");
        int b = sc.nextInt();
        Area rect = new Area(l, b);
        System.out.println("Area of rectangle: " + rect.returnArea());
    }
}
```

```
C:\Windows\System32\cmd.e X + v

Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>java Area.java
Himanshu Saini 24csu350Enter length: 5
Enter breadth: 66
Area of rectangle: 330

C:\Users\ace\OneDrive\Pictures\java>
```

```
File Edit View

import java.util.Scanner;

class Complex {
    int real, imag;

    Complex(int r, int i) {
        real = r;
        imag = i;
    }

    void add(Complex c) {
        System.out.println("Sum: " + (real + c.real) + " + " + (imag + c.imag) + "i");
    }

    void subtract(Complex c) {
        System.out.println("Difference: " + (real - c.real) + " + " + (imag - c.imag) + "i");
    }

    void multiply(Complex c) {
        int r = real * c.real - imag * c.imag;
        int i = real * c.imag + imag * c.real;
        System.out.println("Product: " + r + " + " + i + "i");
    }

    public static void main(String[] args) {
        System.out.print("Himanshu Saini 24csu350");
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter real and imaginary of first number: ");
        Complex c1 = new Complex(sc.nextInt(), sc.nextInt());

        System.out.print("Enter real and imaginary of second number: ");
        Complex c2 = new Complex(sc.nextInt(), sc.nextInt());

        c1.add(c2);
        c1.subtract(c2);
        c1.multiply(c2);
    }
}
```

```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>java Complex.java
Himanshu Saini 24csu350Enter real and imaginary of first number: 4
5
Enter real and imaginary of second number: 5
6
Sum: 9 + 11i
Difference: -1 + -1i
Product: -10 + 49i

C:\Users\ace\OneDrive\Pictures\java>
```

```
import java.util.Scanner;

class Point {
    private float x;
    private float y;

    public Point() {
        this.x = 0.0f;
        this.y = 0.0f;
    }

    public Point(float x, float y) {
        this.x = x;
        this.y = y;
    }

    public float getX() {
        return x;
    }

    public void setX(float x) {
        this.x = x;
    }

    public float getY() {
        return y;
    }

    public void setY(float y) {
        this.y = y;
    }

    public static double distance(Point p1, Point p2) {
        return Math.sqrt(Math.pow(p1.x - p2.x, 2) + Math.pow(p1.y - p2.y, 2));
    }

    public void display() {
        System.out.printf("%7.2f %7.2f\n", x, y);
    }
}
```

```
File Edit View

public static double distance(Point p1, Point p2) {
    return Math.sqrt(Math.pow(p1.x - p2.x, 2) + Math.pow(p1.y - p2.y, 2));
}

public void display() {
    System.out.printf("%7.2f %7.2f\n", x, y);
}

}

public class PointApp {
    public static void main(String[] args) {
        System.out.println("Himanshu Saini, 24csu350");

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter x1: ");
        float x1 = sc.nextFloat();

        System.out.print("Enter y1: ");
        float y1 = sc.nextFloat();

        System.out.print("Enter x2: ");
        float x2 = sc.nextFloat();

        System.out.print("Enter y2: ");
        float y2 = sc.nextFloat();

        Point p1 = new Point(x1, y1);
        Point p2 = new Point(x2, y2);

        System.out.println("Point 1: ");
        p1.display();

        System.out.println("Point 2: ");
        p2.display();

        double dist = Point.distance(p1, p2);
        System.out.printf("Distance between points: %7.2f\n", dist);

        sc.close();
    }
}
```

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java>javac PointApp.java

C:\Users\ace\OneDrive\Pictures\java>java PointApp
Himanshu Saini, 24csu350
Enter x1: 5
Enter y1: 8
Enter x2: 46
Enter y2: 54
Point 1:
    5.00    8.00
Point 2:
    46.00   54.00
Distance between points:    61.62

C:\Users\ace\OneDrive\Pictures\java>
```

Ans1 No.

Ans2 Calling one constructor from another (this() / super()).

Ans3 Constructor without parameters.

Ans4 It becomes a normal method, not a constructor.

Ans5 To restrict object creation (e.g., Singleton).

Ans6 No, only allowed in constructors.

Ans7 Yes, but it becomes a normal method (not a constructor).

Flipped Practicals

Ans1 False

PRACTICAL NO: 5

Student Name and Roll Number: Himanshu Saini and 24csu350
Semester /Section: 3 rd /D
Link to Code:
Date: 17/09/2025
Faculty Signature:

Objective To familiarize the students with the concept of Inheritance.
Program Outcome The students will learn the concept of inheritance.
Program Statement <ol style="list-style-type: none">1. Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both

having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.

2. Create three classes:

Class Vehicle:

Vehicle Class will contain a display() function, which will say "This is a Vehicle".

Class Car:

Car Class will derive the Vehicle Class and overwrite its display() function. it will say "This is a Car".

Class Bike:

Bike Class will derive the Vehicle Class and overwrite its display() function. it will say "This is a Bike".

Write an application that reads an Integer N, which will denote the number of tyres in the vehicle. You have to create an object of the appropriate class according to the value of N and use its display() function.

If N = 2, Create a Bike Object.

If N = 4, Create a Car Object.

Create a Vehicle Object, otherwise.

Definition of Done:

DoD 1: Each class definition is stored in its own .java file.

DoD 2: Switch statement is used for identifying the appropriate class for which the object is

to be invoked.

3. Define a class Box with the following instance variables: width, height and depth, all of type float. Create a new class BoxWeighth that extends Box to include weight as an instance variable. Write an application that tests the functionalities of both these classes.

Definition of Done:

DoD 1: Three java files to be defined. One for each class definition: Box, BoxWeight and BoxWeightDemo.

DoD 2: Box and BoxWeight should have three types of constructors defined: clone of an object, all dimensions specified as arguments, no argument.

DoD 3: Super is used to call base class constructors in derived class

DoD 4: Get and set functions defined as applicable in Box and BoxWeight classes.

DoD 5: Function to display volume in Box class and weight in BoxWeighth class

Background Study

Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance the information is made manageable in a hierarchical order.

The class which inherits the properties of other is known as subclass (derived class, child class) and the class whose properties are inherited is known as superclass (base class, parent class).

extends Keyword

extends is the keyword used to inherit the properties of a class. Following is the syntax of extends keyword.

Syntax

```
class Super {  
    ....  
    ....  
}  
class Sub extends Super {  
    ....
```

```
.....  
}
```

Question Bank

1. What is the use of super keyword?
2. Can an interface be final?
3. Can an abstract class implement an interface?
4. Can you declare an interface method static?

Flipped Questions

Q1. What is the output of the following?

```
class A
```

```
{  
    {  
        System.out.println(1);  
    }  
}
```

```
class B extends A
```

```
{  
    {  
        System.out.println(2);  
    }  
}
```

```
}

class C extends B
{
    {
        System.out.println(3);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();
    }
}
```

Q2. What is the output of the following?

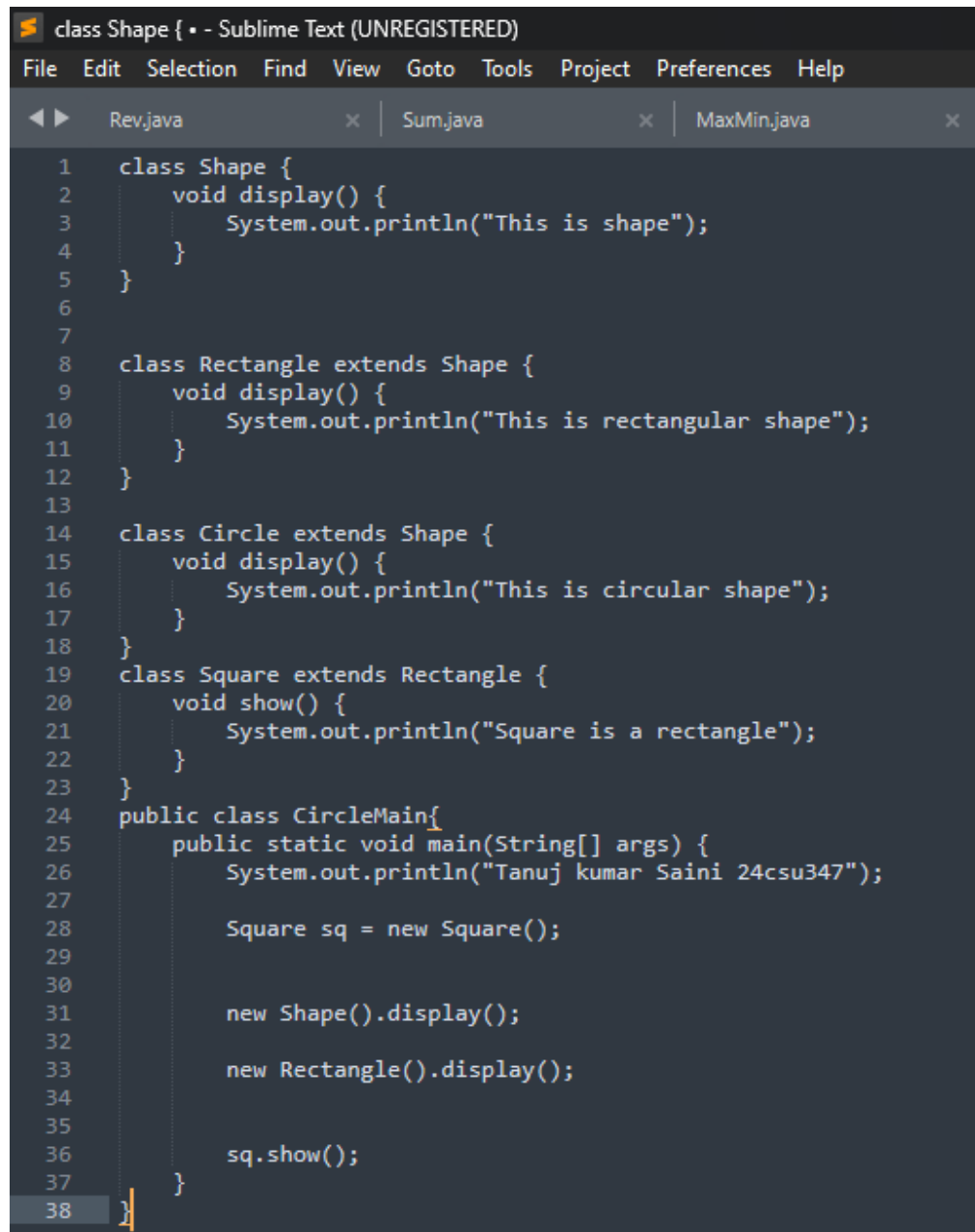
```
class A
{
    public A()
    {
        System.out.println("Class A Constructor");
    }
}
```

```
}  
  
}  
  
class B extends A  
{  
    public B()  
    {  
        System.out.println("Class B Constructor");  
    }  
}  
  
class C extends B  
{  
    public C()  
    {  
        System.out.println("Class C Constructor");  
    }  
}  
  
public class MainClass  
{  
    public static void main(String[] args)  
    {  
        C c = new C();  
    }  
}
```

```
}  
  
}
```

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs



```
class Shape { • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
◀ ▶ Rev.java × Sum.java × MaxMin.java ×

1 class Shape {
2     void display() {
3         System.out.println("This is shape");
4     }
5 }
6
7
8 class Rectangle extends Shape {
9     void display() {
10        System.out.println("This is rectangular shape");
11    }
12 }
13
14 class Circle extends Shape {
15     void display() {
16        System.out.println("This is circular shape");
17    }
18 }
19 class Square extends Rectangle {
20     void show() {
21        System.out.println("Square is a rectangle");
22    }
23 }
24 public class CircleMain{
25     public static void main(String[] args) {
26        System.out.println("Tanuj kumar Saini 24csu347");
27
28        Square sq = new Square();
29
30
31        new Shape().display();
32
33        new Rectangle().display();
34
35
36        sq.show();
37    }
38 }
```



```
C:\Users\ace\OneDrive\Pictures\java>javac CircleMain.java
```

```
C:\Users\ace\OneDrive\Pictures\java>java CircleMain
```

```
Himanshu Saini 24csu350This is shape
```

```
This is rectangular shape
```

```
Square is a rectangle
```

```
C:\Users\ace\OneDrive\Pictures\java2\Vechile.java • - Sublime Text
File Edit Selection Find View Goto Tools Project Prefer

Rev.java x Sum.java x MaxMin.java

1  import java.util.Scanner;
2  class Vehicle {
3      void display() {
4          System.out.println("This is a Vehicle");
5      }
6  }
7
8  class Car extends Vehicle {
9
10     void display() {
11         System.out.println("This is a Car");
12     }
13 }
14 class Bike extends Vehicle {
15
16     void display() {
17         System.out.println("This is a Bike");
18     }
19 }
20 public class VehicleMain {
21     public static void main(String[] args) {
22         System.out.print("Himanshu Saini 24csu350");
23         Scanner sc = new Scanner(System.in);
24
25         System.out.print("Enter number of tyres: ");
26         int N = sc.nextInt();
27
28         Vehicle v;
29
30
31         switch (N) {
32             case 2:
33                 v = new Bike();
34                 break;
35             case 4:
36                 v = new Car();
37                 break;
38             default:
39                 v = new Vehicle();
40                 break;
41         }
42
43         v.display();
44
45         sc.close();
46     }
47 }
48 }
```

Line 22, Column 53

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java2>javac VehicleMain.java

C:\Users\ace\OneDrive\Pictures\java2>java VehicleMain
Himanshu Saini 24csu350Enter number of tyres: 5
This is a Vehicle

C:\Users\ace\OneDrive\Pictures\java2>java VehicleMain
Himanshu Saini 24csu350Enter number of tyres: 2
This is a Bike
```

```
C:\Users\ace\OneDrive\Pictures\java2\WeightMain.java - Sublime Text (U
File Edit Selection Find View Goto Tools Project Preferences
Rev.java x Sum.java x MaxMin.java x
1  import java.util.Scanner;
2  class Shape {
3      void display() {
4          System.out.println("This is shape");
5      }
6  }
7
8  class Rectangle extends Shape {
9      void display() {
10         System.out.println("This is rectangular shape");
11     }
12 }
13
14 class Circle extends Shape {
15     void display() {
16         System.out.println("This is circular shape");
17     }
18 }
19
20 class Square extends Rectangle {
21     void show() {
22         System.out.println("Square is a rectangle");
23     }
24 }
25 class Vehicle {
26     void display() {
27         System.out.println("This is a Vehicle");
28     }
29 }
30
31 class Car extends Vehicle {
32     void display() {
33         System.out.println("This is a Car");
34     }
35 }
36
37 class Bike extends Vehicle {
38     void display() {
39         System.out.println("This is a Bike");
40     }
41 }
42 class Box {
43     private float width, height, depth;
44
45     public Box() {
46         width = height = depth = 0;
47     }
48 }
```

```

C:\Users\ace\OneDrive\Pictures\java2\WeightMain.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Rev.java x Sum.java x MaxMin.java x Search.java

49
50     public Box(float w, float h, float d) {
51         width = w;
52         height = h;
53         depth = d;
54     }
55     public Box(Box b) {
56         this.width = b.width;
57         this.height = b.height;
58         this.depth = b.depth;
59     }
60
61
62     public float getWidth() { return width; }
63     public void setWidth(float width) { this.width = width; }
64
65     public float getHeight() { return height; }
66     public void setHeight(float height) { this.height = height; }
67
68     public float getDepth() { return depth; }
69     public void setDepth(float depth) { this.depth = depth; }
70     public void displayVolume() {
71         float volume = width * height * depth;
72         System.out.println("Volume of Box: " + volume);
73     }
74 }
75
76 class BoxWeight extends Box {
77     private float weight;
78     public BoxWeight() {
79         super();
80         weight = 0;
81     }
82     public BoxWeight(float w, float h, float d, float weight) {
83         super(w, h, d);
84         this.weight = weight;
85     }
86
87
88     public BoxWeight(BoxWeight bw) {
89         super(bw);
90         this.weight = bw.weight;
91     }
92
93
94     public float getWeight() { return weight; }
95     public void setWeight(float weight) { this.weight = weight; }
96

```

```

C:\Users\ace\OneDrive\Pictures\java2\WeightMain.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Rev.java x Sum.java x MaxMin.java x Search.java x
98     public void displayWeight() {
99         System.out.println("Weight of Box: " + weight);
100     }
101 }
102
103 public class WeightMain {
104     public static void main(String[] args) {
105         System.out.print("Himanshu Saini 24csu350");
106         Scanner sc = new Scanner(System.in);
107
108
109
110         System.out.println("\n--- Program 1: Shape Example ---");
111         Square sq = new Square();
112         new Shape().display();
113         new Rectangle().display();
114         sq.show();
115
116         System.out.println("\n--- Program 2: Vehicle Example ---");
117         System.out.print("Enter number of tyres: ");
118         int N = sc.nextInt();
119         Vehicle v;
120         switch (N) {
121             case 2:
122                 v = new Bike();
123                 break;
124             case 4:
125                 v = new Car();
126                 break;
127             default:
128                 v = new Vehicle();
129                 break;
130         }
131         v.display();
132         System.out.println("\n--- Program 3: Box & BoxWeight Example ---");
133         BoxWeight bw1 = new BoxWeight();
134         bw1.setWidth(2);
135         bw1.setHeight(3);
136         bw1.setDepth(4);
137         bw1.setWeight(10);
138         bw1.displayVolume();
139         bw1.displayWeight();
140         BoxWeight bw2 = new BoxWeight(5, 6, 7, 20);
141         bw2.displayVolume();
142         bw2.displayWeight();
143         BoxWeight bw3 = new BoxWeight(bw2);
144         bw3.displayVolume();

```

```
C:\Users\ace\OneDrive\Pictures\java2\WeightMain.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Rev.java x Sum.java x MaxMin.java x Search.java x
126         break;
127     default:
128         v = new Vehicle();
129         break;
130     }
131     v.display();
132     System.out.println("\n--- Program 3: Box & BoxWeight Example ---");
133     BoxWeight bw1 = new BoxWeight();
134     bw1.setWidth(2);
135     bw1.setHeight(3);
136     bw1.setDepth(4);
137     bw1.setWeight(10);
138     bw1.displayVolume();
139     bw1.displayWeight();
140     BoxWeight bw2 = new BoxWeight(5, 6, 7, 20);
141     bw2.displayVolume();
142     bw2.displayWeight();
143     BoxWeight bw3 = new BoxWeight(bw2);
144     bw3.displayVolume();
145     bw3.displayWeight();
146
147     sc.close();
148 }
149 }
```

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ace\OneDrive\Pictures\java2>javac WeightMain.java

C:\Users\ace\OneDrive\Pictures\java2>java WeightMain
Himanshu Saini 24csu350
--- Program 1: Shape Example ---
This is shape
This is rectangular shape
Square is a rectangle

--- Program 2: Vehicle Example ---
Enter number of tyres: 4
This is a Car

--- Program 3: Box & BoxWeight Example ---
Volume of Box: 24.0
Weight of Box: 10.0
Volume of Box: 210.0
Weight of Box: 20.0
Volume of Box: 210.0
Weight of Box: 20.0

C:\Users\ace\OneDrive\Pictures\java2>
```

Ans1. super keyword → used to access parent class variables, methods, and constructors.

Ans2. Interface final? → No, because interfaces are meant to be implemented.

Ans3. Abstract class implement interface? → Yes, but it may leave methods unimplemented.

Ans4. Static method in interface? → No for abstract methods, but from Java 8 static methods with body are allowed.

Flipped Questions

Ans1

OUTPUT: 1

2

3

Ans2

OUTPUT: Class A Constructor

Class B Constructor

Class C Constructor

PRACTICAL NO. 6

Student Name and Roll Number:
Semester /Section:
Link to Code:
Date:
Faculty Signature:

Objective

To familiarize the students with the concept of Interfaces in Java.

Program Outcome

The students will be able to understand where and how interfaces are implemented.

Problem Statement

1. a) Write a program in java to check if a class can extends another class and/ can implement one and more than one interface.

b) Write a program in java to check if an interface can extend other interface.

c) Write a program in java to check if an interface can also extend multiple interfaces.

2. Define the interface / class hierarchy as detailed in the following class diagram

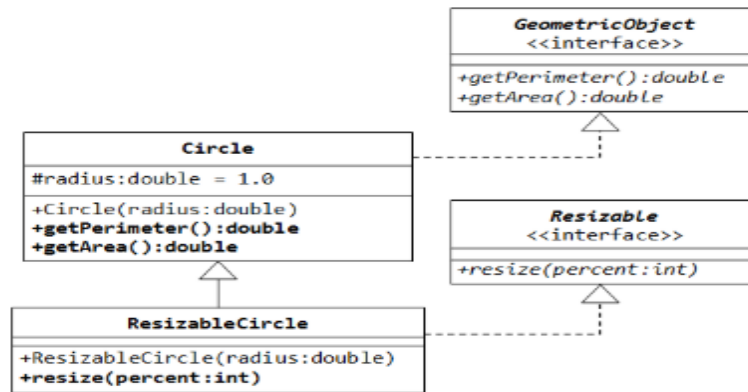
Definition of Done:

DOD 1: The class definitions are defined as per the class diagram.

DOD 2: Each class definition is stored in its own .java file.

DOD 3: Base class constructors are invoked using super keyword

DOD 4: Function overriding is applied wherever applicable.



3. We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.

4. Create an abstract class 'Bank' with an abstract method 'getBalance'. \$100, \$150 and \$200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.

5. We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters

and the marks in four subjects as its parameters for student B. Create an object for each of the two classes and print the percentage of marks for both the students.

Background Study

An **interface in Java** is a blueprint of a class. It has static constants and abstract methods.

The interface in Java is *a mechanism to achieve* abstraction. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java.

In other words, you can say that interfaces can have abstract methods and variables. It cannot have a method body.

Flipped Practicals

1. What is the output of this program?

```
interface calculate {  
    void cal(int item);  
}  
  
class display implements calculate {  
    int x;  
    public void cal(int item) {  
        x = item * item;  
    }  
}
```

```
    }  
}  
  
class interfaces {  
  
    public static void main(String args[]) {  
  
        display arr = new display;  
  
        arr.x = 0;  
  
        arr.cal(2);  
  
        System.out.print(arr.x);  
  
    }  
}
```

- a) 0
- b) 2
- c) 4
- d) None of the mentioned

2. Which of the following package stores all the standard java classes?

- a) lang
- b) java
- c) util
- d) java.packages

3. Determine output of the following code.

```
interface A { }  
  
class C { }
```

```
class D extends C { }

class B extends D implements A { }

public class Test extends Thread{

    public static void main(String[] args){

        B b = new B();

        if (b instanceof A)

            System.out.println("b is an instance of A");

        if (b instanceof C)

            System.out.println("b is an instance of C");

    }

}
```

- a) b is an instance of A.
- b) b is an instance of C.
- c) b is an instance of A followed by b is an instance of C.

Question Bank

1. Can an interface be final?
2. Can an abstract class implement an interface?

3. Can you declare an interface method static?
4. What is the difference between abstract class and interface?
5. When can an object reference can be cast to an interface reference ?
6. Which of the following is true about interfaces in java.

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

Experiment No: 7

Student Name and Roll Number:
Semester /Section:
Link to Code:
Date:
Faculty Signature:

Objective To familiarize the students with package in Java.
Program Outcome The students will learn the concept of package in Java. They will be able to understand
Problem Statement <ol style="list-style-type: none">1. Create a Java package called exercises. Inside the exercises package, create another package (subpackage) called java. Create a Java class called PackageDemo inside the java package. Insert a display() method inside the PackageDemo class. Inside the method, insert this statement: <code>System.out.println("PackageDemo executed");</code> Write a PackageDemoDriver class within the same package to run the display () method of PackageDemo class.2. Write a java program outside the above defined package that imports the package PackageDemo and calls its display() function.
Background Study

A **java package** is a group of similar types of classes, interfaces and sub-packages.

Package in java can be categorized in two form, built-in package and user-defined package.

There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Flipped Practicals

1. What will be the output of the following Java program?

```
package pkg;

class display
{
    int x;

    void show()
    {
        if (x > 1)
            System.out.print(x + " ");
    }
}

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

```
display[] arr=new display[3];

for(int i=0;i<3;i++)

    arr[i]=new display();

arr[0].x = 0;

arr[1].x = 1;

arr[2].x = 2;

for (int i = 0; i < 3; ++i)

    arr[i].show();

}
```

Note : packages.class file is in directory pkg;

- a) 0
- b) 1
- c) 2
- d) 0 1 2

Question Bank

1. What are packages ? what is use of packages ?
2. What is difference between importing "java.applet.Applet" and "java.applet.*" ?
- 3.What do you understand by package access specifier?
4. By default,all program import the java.lang package.
True/False ?

5. Java compiler stores the .class files in the path specified in CLASSPATH environmental variable. True/False ?
6. User-defined package can also be imported just like the standard packages True/False ?
7. A _____ is used to separate the hierarchy of the class while declaring an Import statement.
8. All standard classes of Java are included within a package called _____.

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

PRACTICAL NO.8

Student Name and Roll Number:
Semester /Section:
Link to Code:
Date:
Faculty Signature:

Objective To familiarize the students with exception in java.
Program Outcome The students will learn the concept of exceptions in Java.
Problem Statement <p>1. Write a program that takes 5 integer command line arguments. Create a user defined Exception named CheckArgumentException to check the number of arguments passed through command line. If the number of arguments is less than five, throw the CheckArgumentException, else print the addition of all five numbers.</p> <p>Definition of Done:</p> <p>DOD 1: Create a user-defined exception by the name CheckArgumentException</p> <p>DOD 2: Ask the user to enter the number of arguments</p> <p>DOD 3: Use for loop to enter the arguments</p> <p>DOD 4: Calculate the sum of the values entered</p>

2. Create a class with a `main()` method that *throws* an object of class **Exception** inside a *try* block. Give the constructor for **Exception** a `String` argument. Catch the exception inside a *catch* clause and print the `String` argument. Add a *finally* clause and print a message to prove you were there.
3. Create a three-level hierarchy of exceptions. Now create a base-class A with a method that throws an exception at the base of your hierarchy. Inherit B from A and override the method so it throws an exception at level two of your hierarchy. Repeat by inheriting class C from B. In `main()`, create a C and upcast it to A, then call the method.

Definition of Done:

1. Create three classes A,B,C
 2. Display appropriate messages according to the exceptions.
 3. The program needs to be well commented.
5. Create your own exception class using the `extends` keyword. Write a constructor for this class that takes a `String` argument and stores it inside the object with a `String` reference. Write a method that prints out the stored `String`. Create a *try-catch* clause to exercise your new exception.

Definition of Done:

1. Accept a string argument
2. Create a separate function to print the string.

Background Study

An exception (or exceptional event) is a problem that arises during the execution of a program. When an **Exception** occurs the normal flow of the program is disrupted and the program/Application terminates abnormally, which is not recommended, therefore, these exceptions are to be handled.

An exception can occur for many different reasons. Following are some scenarios where an exception occurs.

- A user has entered an invalid data.
- A file that needs to be opened cannot be found.
- A network connection has been lost in the middle of communications or the JVM has run out of memory.

Question Bank

1. When does Exceptions in Java arises in code sequence?

- a) Run Time
- b) Compilation Time
- c) Can Occur Any Time
- d) None of the mentioned

3. Which of these keywords must be used to monitor for exceptions?

- a) try
- b) finally
- c) throw
- d) catch

4. Which of these keywords must be used to handle the exception thrown by try block in some rational manner?

- a) try
- b) finally
- c) throw
- d) catch

5. Which of these keywords is used to manually throw an exception?

- a) try
- b) finally
- c) throw
- d) catch

Flipped Practicals

1. What will be the output of the following Java program?

```
class exception_handling
{
    public static void main(String args[])
    {
        try
        {
```

```
        System.out.print("Hello" + " " + 1 / 0);  
    }  
    catch(ArithmeticException e)  
    {  
        System.out.print("World");  
    }  
}
```

- a) Hello
- b) World
- c) HelloWorld
- d) Hello World

2. What will be the output of the following Java program?

```
class exception_handling  
{  
    public static void main(String args[])  
    {  
        try  
        {  
            int a, b;  
            b = 0;  
            a = 5 / b;  
            System.out.print("A");  
        }  
        catch(ArithmeticException e)  
        {  
            System.out.print("B");  
        }  
    }  
}
```

- a) A
- b) B
- c) Compilation Error
- d) Runtime Error

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

Student Name and Roll Number:
Semester /Section:
Link to Code:
Date:
Faculty Signature:

PRACTICAL NO. 9

Objective To familiarize the students with File handling.
Program Outcome The students will learn the concept of files in Java.
Problem Statement 1. Write a program to check if the file exist is a file or directory. 2. Write a program to change the file permissions. 3. Write a program to perform simple read and write operation into file. 4. Write a program that writes an Serial No.(int), First Name (String), CGPA (float) and Grade(char) into a text file using bufferedWriter and displays the contents using bufferedReader.

Background Study

File handling is an important part of any application. Java has several methods for creating, reading, updating, and deleting files.

Java File Handling

The File class from the java.io package, allows us to work with files.

To use the File class, create an object of the class, and specify the filename or directory name:

Example

```
import java.io.File; // Import the File class
```

```
File myObj = new File("filename.txt"); // Specify the filename
```

Question Bank

1. Which of these exception is thrown in cases when the file specified for writing is not found?
 - a) IOException
 - b) FileNotFoundException
 - c) FileNotFoundException
 - d) FileInputException
2. Which of these methods are used to read in from file?
 - a) get()
 - b) read()
 - c) scan()
 - d) readfileinput()
3. Which of these values is returned by read() method is end of file (EOF) is encountered?
 - a) 0
 - b) 1
 - c) -1

d) Null

Flipped Practicals

1. What will be the output of the following Java program?

```
1. import java.io.*;
2. class filesinputoutput
3. {
4.     public static void main(String args[])
5.     {
6.         InputStream obj = new FileInputStream("inputoutput.java");
7.         System.out.print(obj.available());
8.     }
9. }
```

Note: inputoutput.java is stored in the disk.

- a) true
- b) false
- c) prints number of bytes in file
- d) prints number of characters in the file

2. What will be the output of the following Java program?

```
1. import java.io.*;
2. public class filesinputoutput
3. {
4.     public static void main(String[] args)
5.     {
6.         String obj = "abc";
7.         byte b[] = obj.getBytes();
8.         ByteArrayInputStream obj1 = new ByteArrayInputStream(b);
9.         for (int i = 0; i < 2; ++ i)
10.        {
11.            int c;
12.            while((c = obj1.read()) != -1)
13.            {
14.                if(i == 0)
15.                {
16.                    System.out.print(Character.toUpperCase((char)c));
17.                    obj2.write(1);
```

```
18.     }  
19.     }  
20.     System.out.print(obj2);  
21.     }  
22.     }  
23. }
```

- a) AaBaCa
- b) ABCaaa
- c) AaaBaaCaa
- d) AaBaaCaaa

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

Annexure 1

Intro to Java

CSL215

Project Report



Faculty name:
name:

Student

Roll No.:

Semester:

Group:

**Department of Computer Science and
Engineering**

**The NorthCap University, Gurugram- 122017,
India**

Session 2025-26

Table of Contents

S.No		Page No.
1.	Project Title:	
2.	Description of Project: Problem Statement	
3.	Problem Analysis 3.1 Hardware Requirements 3.2 Software Requirements	
4.	Design 4.1 Data/Input Output Description: 4.2 Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps	
5.	Implementation and Testing (stage/module wise)	
6.	Output (Screenshots)	
7.	Conclusion and Future Scope	