

Lab-1: Basic Programs

1. Program to Print "Hello, World!"

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

A black background with white text

AI-generated content may be incorrect.

2. Program to Add Two Numbers

import java.util.Scanner;

public class AddTwoNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter first number: ");

int num1 = scanner.nextInt();

System.out.print("Enter second number: ");

int num2 = scanner.nextInt();

int sum = num1 + num2;

System.out.println("Sum: " + sum);

scanner.close();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

3. Program to Calculate Area of a Rectangle

import java.util.Scanner;

public class RectangleArea {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter length: ");

double length = scanner.nextDouble();

System.out.print("Enter width: ");

double width = scanner.nextDouble();

double area = length \* width;

System.out.println("Area of rectangle: " + area);

scanner.close();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

4. Program to Find the Square of a Number

import java.util.Scanner;

public class SquareNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

int num = scanner.nextInt();

int square = num \* num;

System.out.println("Square: " + square);

scanner.close();

}

} A screen shot of a computer program

AI-generated content may be incorrect.

5. Program to Calculate Simple Interest

import java.util.Scanner;

public class SimpleInterest {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter principal amount: ");

double principal = scanner.nextDouble();

System.out.print("Enter rate of interest: ");

double rate = scanner.nextDouble();

System.out.print("Enter time (years): ");

double time = scanner.nextDouble();

double si = (principal \* rate \* time) / 100;

System.out.println("Simple Interest: " + si);

scanner.close();

}

} A black screen with white text

AI-generated content may be incorrect.

6. Program to Convert Celsius to Fahrenheit

import java.util.Scanner;

public class CelsiusToFahrenheit {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter temperature in Celsius: ");

double celsius = scanner.nextDouble();

double fahrenheit = (celsius \* 9 / 5) + 32;

System.out.println("Temperature in Fahrenheit: " + fahrenheit);

scanner.close();

}

}

A computer screen with white text

AI-generated content may be incorrect.

7. Program to Find Maximum of Two Numbers

import java.util.Scanner;

public class MaxOfTwo {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter first number: ");

int num1 = scanner.nextInt();

System.out.print("Enter second number: ");

int num2 = scanner.nextInt();

int max = (num1 > num2) ? num1 : num2;

System.out.println("Maximum: " + max);

scanner.close();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

8. Program to Check if a Number is Even or Odd

import java.util.Scanner;

public class EvenOrOdd {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

int num = scanner.nextInt();

if (num % 2 == 0) {

System.out.println(num + " is even.");

} else {

System.out.println(num + " is odd.");

}

scanner.close();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

9. Swapping of 2 Numbers

import java.util.Scanner;

public class SwapTwoNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter first number: ");

int a = scanner.nextInt();

System.out.print("Enter second number: ");

int b = scanner.nextInt();

System.out.println("Before swapping: a = " + a + ", b = " + b);

int temp = a;

a = b;

b = temp;

System.out.println("After swapping: a = " + a + ", b = " + b);

scanner.close();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

10. Swapping of 3 Numbers

import java.util.Scanner;

public class SwapThreeNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter first number: ");

int a = scanner.nextInt();

System.out.print("Enter second number: ");

int b = scanner.nextInt();

System.out.print("Enter third number: ");

int c = scanner.nextInt();

System.out.println("Before swapping: a = " + a + ", b = " + b + ", c = " + c);

int temp = a;

a = b;

b = c;

c = temp;

System.out.println("After swapping: a = " + a + ", b = " + b + ", c = " + c);

scanner.close();

}

}

A computer screen with numbers and letters

AI-generated content may be incorrect.

11. Data Types Program Using Local and Instance Variables

public class DataTypesExample {

// Instance variables

int instanceInt = 100;

double instanceDouble = 12.34;

char instanceChar = 'A';

boolean instanceBoolean = true;

public static void main(String[] args) {

// Local variables

byte localByte = 10;

short localShort = 500;

long localLong = 100000L;

float localFloat = 5.67f;

String localString = "Hello";

DataTypesExample obj = new DataTypesExample();

System.out.println("Local Variables:");

System.out.println("byte: " + localByte);

System.out.println("short: " + localShort);

System.out.println("long: " + localLong);

System.out.println("float: " + localFloat);

System.out.println("String: " + localString);

System.out.println("\nInstance Variables:");

System.out.println("int: " + obj.instanceInt);

System.out.println("double: " + obj.instanceDouble);

System.out.println("char: " + obj.instanceChar);

System.out.println("boolean: " + obj.instanceBoolean);

}

}

A black screen with white text

AI-generated content may be incorrect.

Lab-2: Operators

1. Arithmetic Operators

public class ArithmeticOperators {

public static void main(String[] args) {

int a = 10, b = 5;

System.out.println("Addition: " + (a + b));

System.out.println("Subtraction: " + (a - b));

System.out.println("Multiplication: " + (a \* b));

System.out.println("Division: " + (a / b));

System.out.println("Modulus: " + (a % b));

}

}

A screen shot of a computer

AI-generated content may be incorrect.

2. Assignment Operators

public class AssignmentOperators {

public static void main(String[] args) {

int a = 10;

System.out.println("Initial a: " + a);

a += 5; // a = a + 5

System.out.println("After += 5: " + a);

a -= 3; // a = a - 3

System.out.println("After -= 3: " + a);

a \*= 2; // a = a \* 2

System.out.println("After \*= 2: " + a);

a /= 4; // a = a / 4

System.out.println("After /= 4: " + a);

}

}

A screen shot of a computer

AI-generated content may be incorrect.

3. Bitwise Operators

public class BitwiseOperators {

public static void main(String[] args) {

int a = 5; // 0101 in binary

int b = 3; // 0011 in binary

System.out.println("AND: " + (a & b)); // 0001

System.out.println("OR: " + (a | b)); // 0111

System.out.println("XOR: " + (a ^ b)); // 0110

System.out.println("Left Shift: " + (a << 1)); // 1010

System.out.println("Right Shift: " + (a >> 1)); // 0010

}

}

A black screen with white text

AI-generated content may be incorrect.

4. Logical Operators

public class LogicalOperators {

public static void main(String[] args) {

boolean a = true, b = false;

System.out.println("AND: " + (a && b));

System.out.println("OR: " + (a || b));

System.out.println("NOT: " + (!a));

}

}

A black screen with white text

AI-generated content may be incorrect.

5. Relational Operators

public class RelationalOperators {

public static void main(String[] args) {

int a = 10, b = 20;

System.out.println("a > b: " + (a > b));

System.out.println("a < b: " + (a < b));

System.out.println("a == b: " + (a == b));

System.out.println("a != b: " + (a != b));

}

}

A screen shot of a computer

AI-generated content may be incorrect.

6. Unary Operator

public class UnaryOperators {

public static void main(String[] args) {

int a = 10;

System.out.println("Initial a: " + a);

a++; // Increment

System.out.println("After a++: " + a);

a--; // Decrement

System.out.println("After a--: " + a);

System.out.println("Negation: " + (-a));

}

}

A screen shot of a computer

AI-generated content may be incorrect.

7. Arithmetic with Instance Variables

public class ArithmeticInstance {

int a = 10;

int b = 5;

public static void main(String[] args) {

ArithmeticInstance obj = new ArithmeticInstance();

System.out.println("Addition: " + (obj.a + obj.b));

System.out.println("Subtraction: " + (obj.a - obj.b));

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

8. Assignment with Instance Variables

public class AssignmentInstance {

int a = 10;

public static void main(String[] args) {

AssignmentInstance obj = new AssignmentInstance();

System.out.println("Initial a: " + obj.a);

obj.a += 5;

System.out.println("After += 5: " + obj.a);

}

}

A screen shot of a computer

AI-generated content may be incorrect.

9. Bitwise with Instance Variables

public class BitwiseInstance {

int a = 5;

int b = 3;

public static void main(String[] args) {

BitwiseInstance obj = new BitwiseInstance();

System.out.println("AND: " + (obj.a & obj.b));

System.out.println("OR: " + (obj.a | obj.b));

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

10. Logical with Instance Variables

public class LogicalInstance {

boolean a = true;

boolean b = false;

public static void main(String[] args) {

LogicalInstance obj = new LogicalInstance();

System.out.println("AND: " + (obj.a && obj.b));

System.out.println("OR: " + (obj.a || obj.b));

}

}

A screen shot of a computer

AI-generated content may be incorrect.

11. Relational with Instance Variables

public class RelationalInstance {

int a = 10;

int b = 20;

public static void main(String[] args) {

RelationalInstance obj = new RelationalInstance();

System.out.println("a < b: " + (obj.a < obj.b));

System.out.println("a == b: " + (obj.a == obj.b));

}

}

A screen shot of a computer

AI-generated content may be incorrect.

12. Unary with Instance Variables

public class UnaryInstance {

int a = 10;

public static void main(String[] args) {

UnaryInstance obj = new UnaryInstance();

System.out.println("Initial a: " + obj.a);

obj.a++;

System.out.println("After a++: " + obj.a);

}

}

A screen shot of a computer

AI-generated content may be incorrect.

13. Sum of Numbers Using Instance Variables

public class SumInstance {

int num1 = 5;

int num2 = 10;

int num3 = 15;

public static void main(String[] args) {

SumInstance obj = new SumInstance();

int sum = obj.num1 + obj.num2 + obj.num3;

System.out.println("Sum: " + sum);

}

}

A screen shot of a computer

AI-generated content may be incorrect.

Lab-3: Control Structures

1. IF Example

public class IfExample {

public static void main(String[] args) {

int num = 10;

if (num > 0) {

System.out.println("Number is positive.");

}

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

2. IF Else Example

public class IfElseExample {

public static void main(String[] args) {

int num = -5;

if (num > 0) {

System.out.println("Number is positive.");

} else {

System.out.println("Number is negative or zero.");

}

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

3. ELSE IF Example

public class ElseIfExample {

public static void main(String[] args) {

int num = 0;

if (num > 0) {

System.out.println("Positive");

} else if (num < 0) {

System.out.println("Negative");

} else {

System.out.println("Zero");

}

}

}

A computer screen with white text

AI-generated content may be incorrect.

4. Else If Ladder Example

import java.util.Scanner;

public class ElseIfLadder {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter marks: ");

int marks = scanner.nextInt();

if (marks >= 90) {

System.out.println("Grade: A");

} else if (marks >= 80) {

System.out.println("Grade: B");

} else if (marks >= 70) {

System.out.println("Grade: C");

} else {

System.out.println("Grade: D");

}

scanner.close();

}

}

A computer screen with white text

AI-generated content may be incorrect.

5. Nested If Example

import java.util.Scanner;

public class NestedIfExample {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter age: ");

int age = scanner.nextInt();

if (age >= 18) {

System.out.print("Do you have a license? (yes/no): ");

String license = scanner.next();

if (license.equalsIgnoreCase("yes")) {

System.out.println("You can drive.");

} else {

System.out.println("You need a license to drive.");

}

} else {

System.out.println("You are too young to drive.");

}

scanner.close();

}

}

A computer screen with white text

AI-generated content may be incorrect.

6. Switch Case Example 1 (Simple Switch Statement)

import java.util.Scanner;

public class SwitchExample1 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a day number (1-7): ");

int day = scanner.nextInt();

switch (day) {

case 1:

System.out.println("Monday");

break;

case 2:

System.out.println("Tuesday");

break;

default:

System.out.println("Invalid day");

}

scanner.close();

}

}

A black background with white text

AI-generated content may be incorrect.

7. Case Example 2 (Switch String Example - Traffic Lights)

import java.util.Scanner;

public class TrafficLights {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter traffic light color: ");

String color = scanner.nextLine().toLowerCase();

switch (color) {

case "red":

System.out.println("Stop");

break;

case "yellow":

System.out.println("Prepare to stop");

break;

case "green":

System.out.println("Go");

break;

default:

System.out.println("Invalid color");

}

scanner.close();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

8. Break Statement

public class BreakExample {

public static void main(String[] args) {

for (int i = 1; i <= 10; i++) {

if (i == 5) {

break;

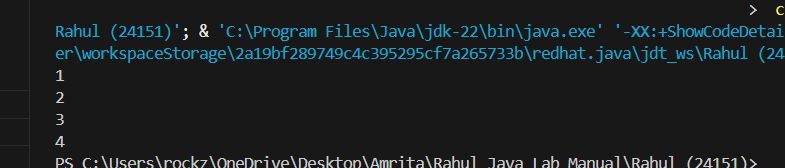
}

System.out.println(i);

}

}

}



9. Continue Statement

public class ContinueExample {

public static void main(String[] args) {

for (int i = 1; i <= 5; i++) {

if (i == 3) {

continue;

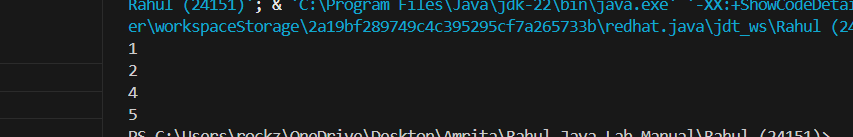
}

System.out.println(i);

}

}

}



10. Do While Loop

public class DoWhileExample {

public static void main(String[] args) {

int i = 1;

do {

System.out.println(i);

i++;

} while (i <= 5);

}

}

A screen shot of a computer

AI-generated content may be incorrect.

11. For Loop

public class ForLoopExample {

public static void main(String[] args) {

for (int i = 1; i <= 5; i++) {

System.out.println(i);

}

}

}

A computer screen shot of a black screen

AI-generated content may be incorrect.

12. While Loop

public class WhileLoopExample {

public static void main(String[] args) {

int i = 1;

while (i <= 5) {

System.out.println(i);

i++;

}

}

}

A computer screen with white text

AI-generated content may be incorrect.

13. Largest Number from Input

import java.util.Scanner;

public class LargestNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter number of elements: ");

int n = scanner.nextInt();

System.out.print("Enter number 1: ");

int max = scanner.nextInt();

for (int i = 2; i <= n; i++) {

System.out.print("Enter number " + i + ": ");

int num = scanner.nextInt();

if (num > max) {

max = num;

}

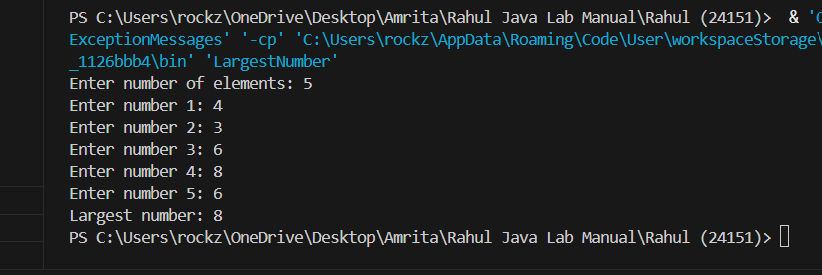
}

System.out.println("Largest number: " + max);

scanner.close();

}

}



14. Sum of Digits of a Given Number

import java.util.Scanner;

public class SumOfDigits {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

int num = scanner.nextInt();

int sum = 0;

while (num > 0) {

sum += num % 10;

num /= 10;

}

System.out.println("Sum of digits: " + sum);

scanner.close();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

15. Reverse a Given Number

import java.util.Scanner;

public class ReverseNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

int num = scanner.nextInt();

int reversed = 0;

while (num > 0) {

reversed = reversed \* 10 + num % 10;

num /= 10;

}

System.out.println("Reversed number: " + reversed);

scanner.close();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

16. Factorial Using Loops

import java.util.Scanner;

public class Factorial {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

int num = scanner.nextInt();

long factorial = 1;

for (int i = 1; i <= num; i++) {

factorial \*= i;

}

System.out.println("Factorial: " + factorial);

scanner.close();

}

}

A black screen with white text

AI-generated content may be incorrect.

17. Fibonacci Series Up to n Terms

import java.util.Scanner;

public class Fibonacci {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter number of terms: ");

int n = scanner.nextInt();

int a = 0, b = 1;

System.out.print("Fibonacci Series: " + a + " " + b);

for (int i = 3; i <= n; i++) {

int next = a + b;

System.out.print(" " + next);

a = b;

b = next;

}

scanner.close();

}

}

A computer screen with white text

AI-generated content may be incorrect.

18. Check if a Number is an Armstrong Number

import java.util.Scanner;

public class ArmstrongNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

int num = scanner.nextInt();

int original = num, sum = 0, digits = String.valueOf(num).length();

while (num > 0) {

int digit = num % 10;

sum += Math.pow(digit, digits);

num /= 10;

}

if (sum == original) {

System.out.println(original + " is an Armstrong number.");

} else {

System.out.println(original + " is not an Armstrong number.");

}

scanner.close();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

19. Check if a Number is a Palindrome

import java.util.Scanner;

public class PalindromeNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

int num = scanner.nextInt();

int original = num, reversed = 0;

while (num > 0) {

reversed = reversed \* 10 + num % 10;

num /= 10;

}

if (original == reversed) {

System.out.println(original + " is a palindrome.");

} else {

System.out.println(original + " is not a palindrome.");

}

scanner.close();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

20. Electricity Bill Calculation

import java.util.Scanner;

public class ElectricityBill {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter units consumed: ");

int units = scanner.nextInt();

double bill = 0;

if (units <= 100) {

bill = units \* 3;

} else if (units <= 300) {

bill = 100 \* 3 + (units - 100) \* 5;

} else {

bill = 100 \* 3 + 200 \* 5 + (units - 300) \* 8;

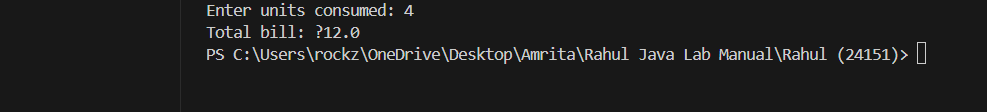
}

System.out.println("Total bill: ₹" + bill);

scanner.close();

}

}



21. Bus Fare Calculation

import java.util.Scanner;

public class BusFare {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter kilometers traveled: ");

int km = scanner.nextInt();

int fare = (km <= 5) ? 10 : 10 + (km - 5) \* 2;

System.out.println("Total fare: ₹" + fare);

scanner.close();

}

}

A black screen with blue and white text

AI-generated content may be incorrect.

22. Movie Ticket Cost

public class MovieTicketCost {

public static void main(String[] args) {

int adults = 5, children = 2;

int adultCost = 100, childCost = 50;

int total = (adults \* adultCost) + (children \* childCost);

System.out.println("Total cost: ₹" + total);

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

23. Shopping Discount Calculation

import java.util.Scanner;

public class ShoppingDiscount {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter purchase amount: ₹");

double amount = scanner.nextDouble();

double finalAmount;

if (amount < 500) {

finalAmount = amount;

} else if (amount <= 2000) {

finalAmount = amount \* 0.9; // 10% discount

} else {

finalAmount = amount \* 0.8; // 20% discount

}

System.out.println("Final bill: ₹" + finalAmount);

scanner.close();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

24. Hospital Bill Calculation

import java.util.Scanner;

public class HospitalBill {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter room type (1-General, 2-Semi-Private, 3-Private): ");

int type = scanner.nextInt();

System.out.print("Enter number of days: ");

int days = scanner.nextInt();

double rate = (type == 1) ? 500 : (type == 2) ? 1000 : 2000;

double bill = rate \* days;

if (days > 7) {

bill \*= 0.95; // 5% discount

}

System.out.println("Final bill: ₹" + bill);

scanner.close();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

25. Grocery Store Bulk Purchase Discount

import java.util.Scanner;

public class GroceryDiscount {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter kg of rice: ");

int rice = scanner.nextInt();

System.out.print("Enter kg of wheat: ");

int wheat = scanner.nextInt();

System.out.print("Enter kg of dry fruits: ");

int dryFruits = scanner.nextInt();

double total = rice \* 40 + wheat \* 50 + dryFruits \* 100;

if (rice + wheat + dryFruits > 10) {

total \*= 0.85; // 15% discount

}

System.out.println("Total cost: ₹" + total);

scanner.close();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

26. Simple Calculator Using Switch

import java.util.Scanner;

public class SimpleCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter first number: ");

double num1 = scanner.nextDouble();

System.out.print("Enter operator (+, -, \*, /): ");

char operator = scanner.next().charAt(0);

System.out.print("Enter second number: ");

double num2 = scanner.nextDouble();

double result;

switch (operator) {

case '+':

result = num1 + num2;

break;

case '-':

result = num1 - num2;

break;

case '\*':

result = num1 \* num2;

break;

case '/':

result = (num2 != 0) ? num1 / num2 : Double.NaN;

break;

default:

System.out.println("Invalid operator");

return;

}

System.out.println("Result: " + result);

scanner.close();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

27. Smart Toll System

import java.util.Scanner;

public class SmartTollSystem {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter vehicle type (1-Car, 2-Truck, 3-Bike): ");

int type = scanner.nextInt();

System.out.print("Enter time of travel (0-23): ");

int time = scanner.nextInt();

System.out.print("Enter number of trips this month: ");

int trips = scanner.nextInt();

System.out.print("Has Fast Tag? (yes/no): ");

String fastTag = scanner.next();

double baseFee = (type == 1) ? 100 : (type == 2) ? 200 : 50;

double cargoWeight = 0;

if (type == 2) {

System.out.print("Enter cargo weight (kg): ");

cargoWeight = scanner.nextDouble();

}

// Time-based adjustments

if ((time >= 8 && time < 11) || (time >= 17 && time < 20)) {

baseFee \*= 1.20; // 20% surcharge

} else if (time >= 23 || time < 5) {

baseFee \*= 0.90; // 10% discount

}

// Frequent traveler discount

if (trips > 10) {

baseFee \*= 0.85; // 15% discount

}

// Fast Tag bonus

if (fastTag.equalsIgnoreCase("yes")) {

baseFee -= 20;

}

// Overloaded truck penalty

if (type == 2 && cargoWeight > 1000) {

baseFee \*= 1.30; // 30% penalty

}

System.out.println("Final toll fee: ₹" + baseFee);

scanner.close();

}

}

A computer screen with text

AI-generated content may be incorrect.

Lab-4: Functions and Classes

1. Reference Type Example (Function)

class Person {

String name;

}

public class ReferenceTypeExample {

public static void changeName(Person p) {

p.name = "Changed";

}

public static void main(String[] args) {

Person person = new Person();

person.name = "Original";

System.out.println("Before: " + person.name);

changeName(person);

System.out.println("After: " + person.name);

}

}

A black screen with white text

AI-generated content may be incorrect.

2. Value Type Example (Function)

public class ValueTypeExample {

public static void changeValue(int x) {

x = 20;

}

public static void main(String[] args) {

int a = 10;

System.out.println("Before: " + a);

changeValue(a);

System.out.println("After: " + a);

}

}

A computer screen with white text

AI-generated content may be incorrect.

3. Object of a Class

class Student {

String name = "John";

int age = 20;

void display() {

System.out.println("Name: " + name + ", Age: " + age);

}

}

public class ObjectExample {

public static void main(String[] args) {

Student student = new Student();

student.display();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

4. Method in the Class to Set Values for Fields

class Employee {

String name;

int id;

void setValues(String n, int i) {

name = n;

id = i;

}

void display() {

System.out.println("ID: " + id + ", Name: " + name);

}

}

public class SetValuesExample {

public static void main(String[] args) {

Employee emp = new Employee();

emp.setValues("Alice", 101);

emp.display();

}

}

A black screen with white text

AI-generated content may be incorrect.

Lab-5: Constructors

1. Default Constructor

class Book {

String title;

Book() {

title = "Unknown";

}

void display() {

System.out.println("Title: " + title);

}

}

public class DefaultConstructor {

public static void main(String[] args) {

Book book = new Book();

book.display();

}

}

A black background with blue text

AI-generated content may be incorrect.

2. Parameterized Constructor

class Car {

String brand;

Car(String b) {

brand = b;

}

void display() {

System.out.println("Brand: " + brand);

}

}

public class ParameterizedConstructor {

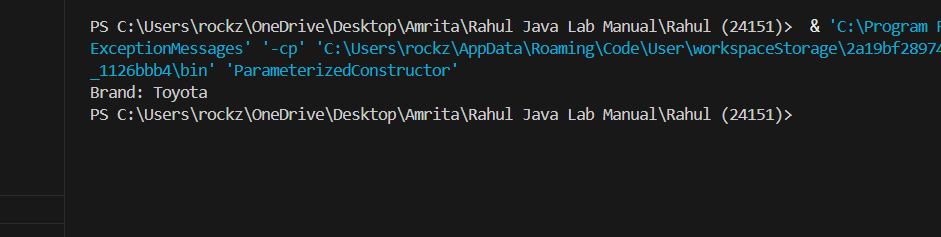
public static void main(String[] args) {

Car car = new Car("Toyota");

car.display();

}

}



3. Constructor Overloading

class Rectangle {

int length, width;

Rectangle() {

length = 0;

width = 0;

}

Rectangle(int l, int w) {

length = l;

width = w;

}

void display() {

System.out.println("Length: " + length + ", Width: " + width);

}

}

public class ConstructorOverloading {

public static void main(String[] args) {

Rectangle rect1 = new Rectangle();

Rectangle rect2 = new Rectangle(5, 3);

rect1.display();

rect2.display();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

4. Garbage Collector (GC)

public class GarbageCollectorExample {

public static void main(String[] args) {

GarbageCollectorExample obj = new GarbageCollectorExample();

obj = null; // Object becomes eligible for GC

System.gc(); // Request garbage collection

System.out.println("Garbage collection requested.");

}

@Override

protected void finalize() {

System.out.println("Object is being garbage collected.");

}

}

A screen shot of a computer

AI-generated content may be incorrect.

5. Constructor Overloading in a Car Class

class Car {

String brand, model;

double price;

Car(String b) {

brand = b;

}

Car(String b, String m) {

brand = b;

model = m;

}

Car(String b, String m, double p) {

brand = b;

model = m;

price = p;

}

void display() {

System.out.println("Brand: " + brand + ", Model: " + model + ", Price: " + price);

}

}

public class CarConstructorOverloading {

public static void main(String[] args) {

Car car1 = new Car("Ford");

Car car2 = new Car("Toyota", "Camry");

Car car3 = new Car("Honda", "Civic", 25000);

car1.display();

car2.display();

car3.display();

}

}

A computer screen shot of a program

AI-generated content may be incorrect.

Lab-6: Methods and Keywords

1. Methods in a Class

class Calculator {

int add(int a, int b) {

return a + b;

}

}

public class MethodsInClass {

public static void main(String[] args) {

Calculator calc = new Calculator();

int sum = calc.add(5, 3);

System.out.println("Sum: " + sum);

}

}

A computer screen with white text

AI-generated content may be incorrect.

2. Method Overloading by Changing Number of Parameters

class MathOperations {

int add(int a, int b) {

return a + b;

}

int add(int a, int b, int c) {

return a + b + c;

}

}

public class MethodOverloadingParams {

public static void main(String[] args) {

MathOperations math = new MathOperations();

System.out.println("Sum of 2 numbers: " + math.add(2, 3));

System.out.println("Sum of 3 numbers: " + math.add(2, 3, 4));

}

}

A computer screen with text

AI-generated content may be incorrect.

3. Method Overloading by Changing Data Types of Parameters

class Printer {

void print(int x) {

System.out.println("Integer: " + x);

}

void print(String x) {

System.out.println("String: " + x);

}

}

public class MethodOverloadingTypes {

public static void main(String[] args) {

Printer printer = new Printer();

printer.print(10);

printer.print("Hello");

}

}

A computer screen with text

AI-generated content may be incorrect.

4. Method Overloading in a Math Operations Class

class MathOperations {

int sum(int a, int b) {

return a + b;

}

double sum(double a, double b) {

return a + b;

}

int sum(int a, int b, int c) {

return a + b + c;

}

}

public class MathOperationsOverloading {

public static void main(String[] args) {

MathOperations math = new MathOperations();

System.out.println("Int sum: " + math.sum(2, 3));

System.out.println("Double sum: " + math.sum(2.5, 3.5));

System.out.println("Three int sum: " + math.sum(1, 2, 3));

}

}

A screen shot of a computer

AI-generated content may be incorrect.

5. Method Overriding in Java

class Parent {

void display() {

System.out.println("Parent class");

}

}

class Child extends Parent {

void display() {

System.out.println("Child class");

}

}

public class MethodOverriding {

public static void main(String[] args) {

Child child = new Child();

child.display();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

6. Method Overriding Using Animal and Dog Classes

class Animal {

void makeSound() {

System.out.println("Some sound");

}

}

class Dog extends Animal {

void makeSound() {

System.out.println("Woof");

}

}

public class AnimalDogOverriding {

public static void main(String[] args) {

Dog dog = new Dog();

dog.makeSound();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

7. Super Keyword (Parent Class Variable)

class Parent {

int x = 10;

}

class Child extends Parent {

int x = 20;

void display() {

System.out.println("Child x: " + x);

System.out.println("Parent x: " + super.x);

}

}

public class SuperVariable {

public static void main(String[] args) {

Child child = new Child();

child.display();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

8. Super Keyword (Parent Class Methods)

class Parent {

void show() {

System.out.println("Parent show");

}

}

class Child extends Parent {

void show() {

super.show();

System.out.println("Child show");

}

}

public class SuperMethod {

public static void main(String[] args) {

Child child = new Child();

child.show();

}

}

A computer screen with white text

AI-generated content may be incorrect.

9. Super Keyword (Parent Class Constructor)

class Parent {

Parent() {

System.out.println("Parent constructor");

}

}

class Child extends Parent {

Child() {

super();

System.out.println("Child constructor");

}

}

public class SuperConstructor {

public static void main(String[] args) {

Child child = new Child();

}

}

A computer screen with text

AI-generated content may be incorrect.

10. Java Final Keyword (Final Variable)

public class FinalVariable {

final int VALUE = 100;

public static void main(String[] args) {

FinalVariable obj = new FinalVariable();

System.out.println("Final value: " + obj.VALUE);

// obj.VALUE = 200; // This would cause a compilation error

}

}

A computer screen with white text

AI-generated content may be incorrect.

11. Java Final Keyword (Final Method)

class Parent {

final void display() {

System.out.println("This is a final method");

}

}

class Child extends Parent {

// Cannot override display()

}

public class FinalMethod {

public static void main(String[] args) {

Child child = new Child();

child.display();

}

}

A computer screen with white text

AI-generated content may be incorrect.

12. Java Final Keyword (Final Class)

final class FinalClass {

void show() {

System.out.println("This is a final class");

}

}

// class SubClass extends FinalClass {} // This would cause an error

public class FinalClassExample {

public static void main(String[] args) {

FinalClass obj = new FinalClass();

obj.show();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

Lab-7: Inheritance and Interfaces

1. Single Inheritance

class Animal {

void eat() {

System.out.println("Eating");

}

}

class Dog extends Animal {

void bark() {

System.out.println("Barking");

}

}

public class SingleInheritance {

public static void main(String[] args) {

Dog dog = new Dog();

dog.eat();

dog.bark();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

2. Multilevel Inheritance

class Animal {

void eat() {

System.out.println("Eating");

}

}

class Mammal extends Animal {

void walk() {

System.out.println("Walking");

}

}

class Dog extends Mammal {

void bark() {

System.out.println("Barking");

}

}

public class MultilevelInheritance {

public static void main(String[] args) {

Dog dog = new Dog();

dog.eat();

dog.walk();

dog.bark();

}

}

A computer screen with white text

AI-generated content may be incorrect.

3. Hierarchical Inheritance

class Animal {

void eat() {

System.out.println("Eating");

}

}

class Dog extends Animal {

void bark() {

System.out.println("Barking");

}

}

class Cat extends Animal {

void meow() {

System.out.println("Meowing");

}

}

public class HierarchicalInheritance {

public static void main(String[] args) {

Dog dog = new Dog();

Cat cat = new Cat();

dog.eat();

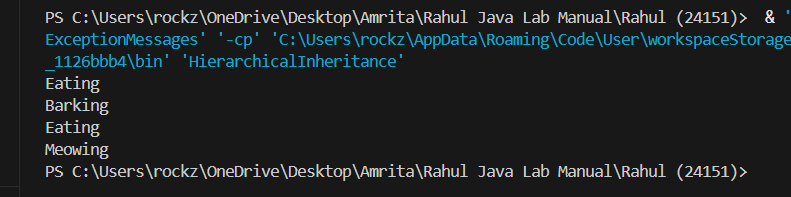
dog.bark();

cat.eat();

cat.meow();

}

}



4. Multiple Inheritance (Using Interfaces)

interface Flyable {

void fly();

}

interface Swimmable {

void swim();

}

class Duck implements Flyable, Swimmable {

public void fly() {

System.out.println("Flying");

}

public void swim() {

System.out.println("Swimming");

}

}

public class MultipleInheritance {

public static void main(String[] args) {

Duck duck = new Duck();

duck.fly();

duck.swim();

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

5. Hybrid Inheritance (Using Interfaces)

interface Animal {

void eat();

}

class Bird implements Animal {

public void eat() {

System.out.println("Eating");

}

}

interface Flyable {

void fly();

}

class Sparrow extends Bird implements Flyable {

public void fly() {

System.out.println("Flying");

}

}

public class HybridInheritance {

public static void main(String[] args) {

Sparrow sparrow = new Sparrow();

sparrow.eat();

sparrow.fly();

}

}

A computer screen with white text

AI-generated content may be incorrect.

6. Implementation of Interface in Java

interface Shape {

double area();

}

class Circle implements Shape {

double radius;

Circle(double r) {

radius = r;

}

public double area() {

return Math.PI \* radius \* radius;

}

}

public class InterfaceExample {

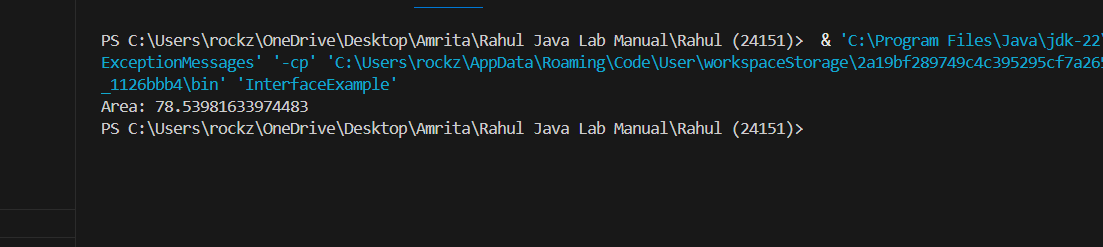
public static void main(String[] args) {

Circle circle = new Circle(5);

System.out.println("Area: " + circle.area());

}

}



7. Relationship Between Class and Interface

interface Printable {

void print();

}

class Document implements Printable {

String content;

Document(String c) {

content = c;

}

public void print() {

System.out.println("Content: " + content);

}

}

public class ClassInterfaceRelationship {

public static void main(String[] args) {

Document doc = new Document("Hello World");

doc.print();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

Lab-8: Packages and Exception Handling

1. Packages

// File: mypackage/MyClass.java

package mypackage;

public class MyClass {

public void display() {

System.out.println("Hello from MyClass");

}

}

// File: Main.java

import mypackage.MyClass;

public class Main {

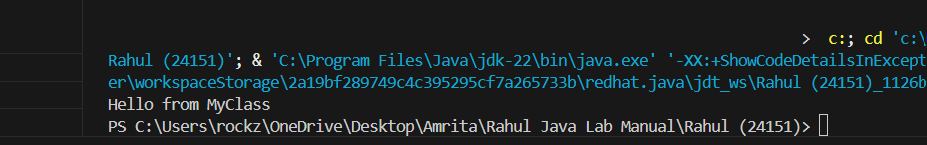
public static void main(String[] args) {

MyClass obj = new MyClass();

obj.display();

}

}



2. Sub Packages

// File: mypackage.subpackage/SubClass.java

package mypackage.subpackage;

public class SubClass {

public void show() {

System.out.println("Hello from SubClass");

}

}

// File: Main.java

import mypackage.subpackage.SubClass;

public class Main {

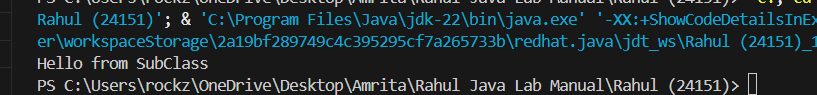
public static void main(String[] args) {

SubClass obj = new SubClass();

obj.show();

}

}



3. Handling a Built-in Exception

public class BuiltInException {

public static void main(String[] args) {

try {

int result = 10 / 0;

System.out.println(result);

} catch (ArithmeticException e) {

System.out.println("Error: Division by zero");

}

}

}

A screen shot of a computer

AI-generated content may be incorrect.

4. Creation of Custom Exception

class CustomException extends Exception {

CustomException(String message) {

super(message);

}

}

public class CustomExceptionExample {

public static void main(String[] args) {

try {

throw new CustomException("This is a custom exception");

} catch (CustomException e) {

System.out.println(e.getMessage());

}

}

}

A screen shot of a computer

AI-generated content may be incorrect.

5. Using throw to Manually Throw an Exception

public class ThrowExample {

public static void main(String[] args) {

try {

throw new Exception("Manual exception");

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

}

A screenshot of a computer

AI-generated content may be incorrect.

6. Java Program Using throws Arithmetic Exception

public class ThrowsExample {

static void divide(int a, int b) throws ArithmeticException {

int result = a / b;

System.out.println("Result: " + result);

}

public static void main(String[] args) {

try {

divide(10, 0);

} catch (ArithmeticException e) {

System.out.println("Error: Division by zero");

}

}

}

A screen shot of a computer

AI-generated content may be incorrect.

7. Invalid Age Exception

class InvalidAgeException extends Exception {

InvalidAgeException(String message) {

super(message);

}

}

public class InvalidAge {

static void checkAge(int age) throws InvalidAgeException {

if (age < 18) {

throw new InvalidAgeException("Age must be 18 or older");

}

System.out.println("Age is valid");

}

public static void main(String[] args) {

try {

checkAge(16);

} catch (InvalidAgeException e) {

System.out.println(e.getMessage());

}

}

}

A screen shot of a computer

AI-generated content may be incorrect.

8. Insufficient Balance Exception

class InsufficientBalanceException extends Exception {

InsufficientBalanceException(String message) {

super(message);

}

}

public class InsufficientBalance {

static void withdraw(double balance, double amount) throws InsufficientBalanceException {

if (amount > balance) {

throw new InsufficientBalanceException("Insufficient balance");

}

System.out.println("Withdrawal successful");

}

public static void main(String[] args) {

try {

withdraw(100, 150);

} catch (InsufficientBalanceException e) {

System.out.println(e.getMessage());

}

}

}

A screen shot of a computer program

AI-generated content may be incorrect.

9. Invalid Marks Exception

class InvalidMarksException extends Exception {

InvalidMarksException(String message) {

super(message);

}

}

public class InvalidMarks {

static void validateMarks(int marks) throws InvalidMarksException {

if (marks < 0 || marks > 100) {

throw new InvalidMarksException("Marks must be between 0 and 100");

}

System.out.println("Marks are valid");

}

public static void main(String[] args) {

try {

validateMarks(120);

} catch (InvalidMarksException e) {

System.out.println(e.getMessage());

}

}

}

A computer screen with white text

AI-generated content may be incorrect.

10. Product Out of Stock Exception

class ProductOutOfStockException extends Exception {

ProductOutOfStockException(String message) {

super(message);

}

}

public class ProductOutOfStock {

static void checkStock(int stock) throws ProductOutOfStockException {

if (stock <= 0) {

throw new ProductOutOfStockException("Product out of stock");

}

System.out.println("Product available");

}

public static void main(String[] args) {

try {

checkStock(0);

} catch (ProductOutOfStockException e) {

System.out.println(e.getMessage());

}

}

}

A screen shot of a computer

AI-generated content may be incorrect.

11. Invalid Username Exception

class InvalidUsernameException extends Exception {

InvalidUsernameException(String message) {

super(message);

}

}

public class InvalidUsername {

static void validateUsername(String username) throws InvalidUsernameException {

if (username.length() < 5) {

throw new InvalidUsernameException("Username must be at least 5 characters");

}

System.out.println("Username is valid");

}

public static void main(String[] args) {

try {

validateUsername("abc");

} catch (InvalidUsernameException e) {

System.out.println(e.getMessage());

}

}

}

A screenshot of a computer program

AI-generated content may be incorrect.

Lab-9: Multithreading

1. Print Multiplication Table Using Thread

class MultiplicationTable extends Thread {

public void run() {

for (int i = 1; i <= 10; i++) {

System.out.println("5 \* " + i + " = " + (5 \* i));

}

}

}

public class MultiplicationThread {

public static void main(String[] args) {

MultiplicationTable table = new MultiplicationTable();

table.start();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

2. Even and Odd Numbers Printing Using Two Threads

class NumberPrinter {

boolean isEven = false;

synchronized void printEven(int num) {

while (!isEven) {

try {

wait();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println("Even: " + num);

isEven = false;

notify();

}

synchronized void printOdd(int num) {

while (isEven) {

try {

wait();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println("Odd: " + num);

isEven = true;

notify();

}

}

public class EvenOddThreads {

public static void main(String[] args) {

NumberPrinter printer = new NumberPrinter();

Thread evenThread = new Thread(() -> {

for (int i = 2; i <= 10; i += 2) {

printer.printEven(i);

}

});

Thread oddThread = new Thread(() -> {

for (int i = 1; i <= 9; i += 2) {

printer.printOdd(i);

}

});

evenThread.start();

oddThread.start();

}

}

A screen shot of a computer

AI-generated content may be incorrect.

3. Factorial Using Thread

class FactorialThread extends Thread {

int num;

FactorialThread(int n) {

num = n;

}

public void run() {

long factorial = 1;

for (int i = 1; i <= num; i++) {

factorial \*= i;

}

System.out.println("Factorial of " + num + ": " + factorial);

}

}

public class FactorialUsingThread {

public static void main(String[] args) {

FactorialThread thread = new FactorialThread(5);

thread.start();

}

}

A screenshot of a computer

AI-generated content may be incorrect.

4. Fibonacci Series Using Thread

class FibonacciThread extends Thread {

int n;

FibonacciThread(int terms) {

n = terms;

}

public void run() {

int a = 0, b = 1;

System.out.print("Fibonacci Series: " + a + " " + b);

for (int i = 3; i <= n; i++) {

int next = a + b;

System.out.print(" " + next);

a = b;

b = next;

}

}

}

public class FibonacciUsingThread {

public static void main(String[] args) {

FibonacciThread thread = new FibonacciThread(5);

thread.start();

}

}

A computer screen shot of a computer program

AI-generated content may be incorrect.