**Chapter 2**

**Self-Review Exercises**

**2.1 Fill in the blanks in each of the following statements:**

a) A(n) **opening brace {** begins the body of every method, and a(n) **closing brace }** ends the body of every method.

b) You can use the **if** statement to make decisions.

c) **//** begins an end-of-line comment.

d) **Spaces, tabs, and newlines** are called white space.

e)  **Keywords** are reserved for use by Java.

f) Java applications begin execution at method **main**.

g) Methods **print** , **println**, and **printf** display information in a command window .

**2.2 State whether each of the following is true or false. If false, explain why.**

a) Comments cause the computer to print the text after the // on the screen when the program executes.

***ANSWER***

**False.**

**Comments do not cause the computer to print anything on the screen.  
They are ignored by the compiler and are meant only for programmers to read and understand the code.**

b) All variables must be given a type when they’re declared.

***ANSWER***

**True.**

**Java is a strongly typed language, which means every variable’s type must be known at compile time — before the program runs.**

c) Java considers the variables number and NuMbEr to be identical.

***ANSWER***

**False.**

**Java is case-sensitive, which means it treats uppercase and lowercase letters as different.**

d) The remainder operator (%) can be used only with integer operands.

***ANSWER***

**False.**

**In Java, the remainder operator (%) can be used with both integer and floating-point operands.**

e) The arithmetic operators \*, /, %, + and - all have the same level of precedence.

***ANSWER***

**False.**

**In Java, the arithmetic operators do not all have the same level of precedence.**

**Here’s the correct order (from highest to lowest):**

1. **\*, /, % → higher precedence**
2. **+, - → lower precedence**

**So multiplication, division, and remainder are evaluated before addition and subtraction.**

**2.3 Write statements to accomplish each of the following tasks:**

a) Declare variables c, thisIsAVariable, q76354 and number to be of type int.

***ANSWER***

**int c;**

**int thisIsAVariable ;**

**int q76354 ;**

**int number;**

b) Prompt the user to enter an integer.

***ANSWER***

**import java.util.Scanner; // import Scanner class**

**public class Main {**

**public static void main(String[] args) {**

**Scanner input = new Scanner(System.in); // create a Scanner object**

**System.out.print("Enter an integer: "); // prompt the user**

**int number = input.nextInt(); // read an integer from the user**

**System.out.println("You entered: " + number); // display the input**

**}**

**}**

c) Input an integer and assign the result to int variable value. Assume Scanner variable input can be used to read a value from the keyboard.

***ANSWER***

**import java.util.Scanner; // import Scanner class**

**public class Main {**

**public static void main(String[] args) {**

**Scanner input = new Scanner(System.in); // create a Scanner object**

**System.out.print("Enter an integer: "); // prompt the user**

**int value = input.nextInt(); // read an integer and assign to variable 'value'**

**System.out.println("You entered: " + value); // display the input**

**}**

**}**

d) Print "This is a Java program" on one line in the command window. Use method System.out.println.

***ANSWER***

**public class Main {**

**public static void main(String[] args) {**

**System.out.println("This is a Java program");**

**}**

**}**

e) Print "This is a Java program" on two lines in the command window. The first line should end with Java. Use method System.out.printf and two %s format specifiers.

***ANSWER***

**public class Main {**

**public static void main(String[] args) {**

**System.out.printf("%s %s%n%s%n", "This is a", "Java", "program");**

**}**

**}**

f) If the variable number is not equal to 7, display "The variable number is not equal to 7”

***ANSWER***

**int number = 5; // example value**

**if (number != 7) {**

**System.out.println("The variable number is not equal to 7");**

**}**

**2.4 Identify and correct the errors in each of the following statements:**

a) if (c < 7); System.out.println("c is less than 7");

***ANSWER***

**if (c < 7) {**

**System.out.println("c is less than 7");**

**}**

b) if (c => 7) System.out.println("c is equal to or greater than 7");

***ANSWER***

**if (c >= 7)**

**System.out.println("c is equal to or greater than 7");**

**2.5 Write declarations, statements or comments that accomplish each of the following tasks:**

a) State that a program will calculate the product of three integers.

ANSWER

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

// This program will calculate the product of three integers

Scanner input = new Scanner(System.in);

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

int a = input.nextInt();

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

int b = input.nextInt();

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

int c = input.nextInt();

int product = a \* b \* c;

System.out.println("The product is: " + product);

}

}

b) Create a Scanner called input that reads values from the standard input.

ANSWER

import java.util.Scanner; // Import the Scanner class

public class Main {

public static void main(String[] args) {

// Create a Scanner called input that reads values from the standard input

Scanner input = new Scanner(System.in);

System.out.print("Enter a number: "); // Prompt the user

int number = input.nextInt(); // Read an integer from the user

System.out.println("You entered: " + number); // Display the entered value

}

}

c) Declare the variables x, y, z and result to be of type int.

ANSWER

int x, y, z, result;

d) Prompt the user to enter the first integer.

ANSWER

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner input = new Scanner(System.in); // create Scanner

System.out.print("Enter the first integer: "); // prompt user

int x = input.nextInt(); // read the integer

}

}

e) Read the first integer from the user and store it in the variable x.

ANSWER

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner input = new Scanner(System.in); // create Scanner

System.out.print("Enter the first integer: "); // prompt user

int x = input.nextInt(); // read integer and store in x

System.out.println("You entered: " + x); // display the value

}

}

f) Prompt the user to enter the second integer.

**g) Read the second integer from the user and store it in y**

System.out.print("Enter the second integer: ");

int y = input.nextInt();

**h) Prompt the user to enter the third integer**

System.out.print("Enter the third integer: ");

**i) Read the third integer from the user and store it in z**

int z = input.nextInt();

**j) Compute the product of the three integers and store in result**

int result = x \* y \* z;

**k) Display "Product is" followed by the value using System.out.printf**

System.out.printf("Product is %d%n", result);

**2.6 Using the statements you wrote in Exercise 2.5, write a complete program that calculates and prints the product of three integers.**

ANSWER

import java.util.Scanner;

public class ProductThreeIntegers {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Prompt and read three integers

System.out.print("Enter the first integer: ");

int x = input.nextInt();

System.out.print("Enter the second integer: ");

int y = input.nextInt();

System.out.print("Enter the third integer: ");

int z = input.nextInt();

// Compute product

int result = x \* y \* z;

// Display result

System.out.printf("Product is %d%n", result);

}

}

**Exercises**

**2.7 Fill in the blanks in each of the following statements:**

a) **Comments** are used to document a program and improve its readability.  
b) A decision can be made in a Java program with a(n) **if statement**.  
c) Calculations are normally performed by **arithmetic/statements**.  
d) The arithmetic operators with the same precedence as multiplication are **/** and **%**.  
e) When parentheses in an arithmetic expression are nested, the **innermost** set of parentheses is evaluated first.  
f) A location in the computer’s memory that may contain different values at various times throughout the execution of a program is called a(n) **variable**.

**2.8 Write Java statements that accomplish each of the following tasks**: **a) Display the message "Enter an integer: ", leaving the cursor on the same line:**

System.out.print("Enter an integer: ");

✅ **Explanation:**

* System.out.print() prints text **without moving to a new line**, so the cursor stays on the same line.

**b) Assign the product of variables b and c to variable a:**

int a = b \* c;

✅ **Explanation:**

* Multiplies b and c and stores the result in a.

**c) Use a comment to state that a program performs a sample payroll calculation:**

// This program performs a sample payroll calculation

✅ **Explanation:**

* Comments starting with // are ignored by Java.
* They are used to explain what the program does for rea

to state that a program performs a sample payroll calculation.

**2.9 State whether each of the following is true or false. If false, explain why.**

a) Java operators are evaluated from left to right.

b) The following are all valid variable names: \_under\_bar\_, m928134, t5, j7, her\_sales$, his\_$account\_total, a, b$, c, z and z2.

c) A valid Java arithmetic expression with no parentheses is evaluated from left to right.

d) The following are all invalid variable names: 3g, 87, 67h2, h22 and 2h.

**2.10 Assuming that x=2 and y=3, what does each of the following statements display?**

**a) Java operators are evaluated from left to right.**  
❌ **False**

* **Explanation:** Operators in Java are evaluated based on **operator precedence**, not strictly left to right.
* For example: 3 + 4 \* 2 → multiplication happens first because \* has higher precedence than +.

**b) The following are all valid variable names:**  
\_under\_bar\_, m928134, t5, j7, her\_sales$, his\_$account\_total, a, b$, c, z, z2  
✅ **True**

* **Explanation:** All names follow Java rules:
  + Start with a letter, $, or \_.
  + Can contain letters, digits, $, or \_.
  + Not a Java keyword.

**c) A valid Java arithmetic expression with no parentheses is evaluated from left to right.**  
❌ **False**

* **Explanation:** Expressions are evaluated based on **operator precedence and associativity**, not strictly left to right.
* Example: 2 + 3 \* 4 → multiplication happens first (3 \* 4), then addition.

**d) The following are all invalid variable names:**  
3g, 87, 67h2, h22, 2h  
❌ **False**

* **Explanation:** h22 is a **valid variable name**.
* The others are invalid because **variable names cannot start with a digit**.

2.11 Which of the following Java statements contain variables whose values are modified?

**a) p = i + j + k + 7;** ✅ **Yes**

* **Explanation:** The variable p is assigned a new value. Its value is **modified**.

**b) System.out.println("variables whose values are modified");** ❌ **No**

* **Explanation:** This only prints text to the console. No variables are changed.

**c) System.out.println("a = 5");** ❌ **No**

* **Explanation:** This prints a string. It does **not assign** the value 5 to a variable.

**d) value = input.nextInt();** ✅ **Yes**

* **Explanation:** The variable value is assigned a new value read from the user. Its value is **modified**.

**2.12 Given that y = ax3 + 7, which of the following are correct Java statements for this equation?**

**a) y = a \* x \* x \* x + 7;** ✅ **Correct**

* Explanation: a \* x \* x \* x computes a·x³, then + 7 is correct.

**b) y = a \* x \* x \* (x + 7);** ❌ **Incorrect**

* Explanation: Expands to a \* x² \* (x + 7) = a·x³ + 7·a·x → **not the same**.

**c) y = (a \* x) \* x \* (x + 7);** ❌ **Incorrect**

* Explanation: Expands to (a\*x) \* x \* (x+7) = a·x³ + 7·a·x² → **not correct**.

**d) y = (a \* x) \* x \* x + 7;** ✅ **Correct**

* Explanation: (a\*x) \* x \* x = a·x³, then +7 → correct.

**e) y = a \* (x \* x \* x) + 7;** ✅ **Correct**

* Explanation: Explicitly calculates x³, multiplies by a, then adds 7 → correct.

**f) y = a \* x \* (x \* x + 7);** ❌ **Incorrect**

✅ **Correct Java statements:** **a, d, e**

**2.13 State the order of evaluation of the operators in each of the following Java statements, and show the value of x after each statement is performed:**

**a) x = 7 + 3 \* 6 / 2 - 1;**

**Operator precedence:**

1. \* and / → evaluated left to right
2. + and - → evaluated left to right

**Step-by-step:**

1. 3 \* 6 = 18
2. 18 / 2 = 9
3. 7 + 9 = 16
4. 16 - 1 = 15

✅ **Result:** x = 15

**b) x = 2 % 2 + 2 \* 2 - 2 / 2;**

**Operator precedence:**

1. %, \*, / → left to right
2. +, - → left to right

**Step-by-step:**

1. 2 % 2 = 0
2. 2 \* 2 = 4
3. 2 / 2 = 1
4. Combine: 0 + 4 - 1 = 3

✅ **Result:** x = 3

**c) x = (3 \* 9 \* (3 + (9 \* 3 / (3))));**

**Operator precedence:**

1. Innermost parentheses first
2. \* and / from left to right

**Step-by-step:**

1. Innermost: 9 \* 3 = 27
2. 27 / 3 = 9
3. Next: 3 + 9 = 12
4. Multiply outer: 3 \* 9 = 27
5. 27 \* 12 = 324

✅ **Result:** x = 324

So final values:

* **a)** 15
* **b)** 3
* **c)** 324

**2.14 Write an application that displays the numbers 1 to 4 on the same line, with each pair of adjacent numbers separated by one space. Use the following techniques:**

**a) Using one System.out.println statement**

public class NumbersDisplay {

public static void main(String[] args) {

System.out.println("1 2 3 4");

}

}

✅ **Explanation:**

* Prints the string "1 2 3 4" on one line.

**b) Using four System.out.print statements**

public class NumbersDisplay {

public static void main(String[] args) {

System.out.print("1 ");

System.out.print("2 ");

System.out.print("3 ");

System.out.print("4");

}

}

✅ **Explanation:**

* Each number is printed without moving to a new line (print not println).
* Spaces are added manually.

**c) Using one System.out.printf statement**

public class NumbersDisplay {

public static void main(String[] args) {

System.out.printf("%d %d %d %d%n", 1, 2, 3, 4);

}

}

✅ **Explanation:**

* %d → format specifier for integers.
* %n → moves to the next line after printing.
* Prints all numbers with spaces in between.

www.EBooksWorld.ir 68 Chapter 2 Introduction to Java Applications; Input/Output and Operators

**2.15 Arithmetic: Sum, Product, Difference, Quotient of Two Integers**

import java.util.Scanner;

public class ArithmeticOperations {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the first integer: ");

int num1 = input.nextInt();

System.out.print("Enter the second integer: ");

int num2 = input.nextInt();

int sum = num1 + num2;

int product = num1 \* num2;

int difference = num1 - num2;

int quotient = num1 / num2; // integer division

System.out.printf("Sum: %d%n", sum);

System.out.printf("Product: %d%n", product);

System.out.printf("Difference: %d%n", difference);

System.out.printf("Quotient: %d%n", quotient);

}

}

**2.16 Comparing Integers**

import java.util.Scanner;

public class CompareIntegers {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the first integer: ");

int num1 = input.nextInt();

System.out.print("Enter the second integer: ");

int num2 = input.nextInt();

if (num1 > num2) {

System.out.println(num1 + " is larger");

} else if (num2 > num1) {

System.out.println(num2 + " is larger");

} else {

System.out.println("These numbers are equal");

}

}

}

**2.17 Arithmetic, Smallest and Largest of Three Integers**

import java.util.Scanner;

public class ThreeIntegersStats {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int x = input.nextInt();

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

int y = input.nextInt();

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

int z = input.nextInt();

int sum = x + y + z;

int average = sum / 3; // integer division

int product = x \* y \* z;

// Determine smallest

int smallest = x;

if (y < smallest) smallest = y;

if (z < smallest) smallest = z;

// Determine largest

int largest = x;

if (y > largest) largest = y;

if (z > largest) largest = z;

System.out.printf("Sum: %d%n", sum);

System.out.printf("Average: %d%n", average);

System.out.printf("Product: %d%n", product);

System.out.printf("Smallest: %d%n", smallest);

System.out.printf("Largest: %d%n", largest);

}

}

**2.18 Displaying Shapes with Asterisks**

Example program to display a **box, oval, arrow, and diamond** using System.out.println:

public class Shapes {

public static void main(String[] args) {

// Box

System.out.println("\*\*\*\*\*");

System.out.println("\* \*");

System.out.println("\* \*");

System.out.println("\*\*\*\*\*\n");

// Oval

System.out.println(" \*\*\* ");

System.out.println("\* \*");

System.out.println("\* \*");

System.out.println(" \*\*\*\n");

// Arrow

System.out.println(" \* ");

System.out.println(" \*\*\* ");

System.out.println("\*\*\*\*\*");

System.out.println(" \* ");

System.out.println(" \* \n");

// Diamond

System.out.println(" \* ");

System.out.println(" \*\*\* ");

System.out.println("\*\*\*\*\*");

System.out.println(" \*\*\* ");

System.out.println(" \* ");

}

}

**2.19 What does this code print?**

System.out.printf("\*%n\*\*%n\*\*\*%n\*\*\*\*%n\*\*\*\*\*%n");

**Output:**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

✅ %n moves to the next line after each string.

**2.20 What does this code print?**

System.out.println("\*");

System.out.println("\*\*\*");

System.out.println("\*\*\*\*\*");

System.out.println("\*\*\*\*");

System.out.println("\*\*");

**Output:**

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*

**2.21 What does this code print?**

System.out.print("\*");

System.out.print("\*\*\*");

System.out.print("\*\*\*\*\*");

System.out.print("\*\*\*\*");

System.out.println("\*\*");

**Output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*

* Everything prints on **one line** because print does not add a newline.
* Only the last println("\*\*") adds a newline at the end.

**2.22 What does this code print?**

System.out.print("\*");

System.out.println("\*\*\*");

System.out.println("\*\*\*\*\*");

System.out.print("\*\*\*\*");

System.out.println("\*\*");

**Output:**

\*\*\*\*

\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*

* Step by step:
  1. print("\*") → \* (no newline)
  2. println("\*\*\*") → adds \*\*\* → line now: \*\*\*\*
  3. println("\*\*\*\*\*") → next line: \*\*\*\*\*
  4. print("\*\*\*\*") → \*\*\*\* on next line (no newline)
  5. println("\*\*") → \*\* → final line: \*\*\*\*\*\*

**2.23 What does this code print?**

System.out.printf("%s%n%s%n%s%n", "\*", "\*\*\*", "\*\*\*\*\*");

**Output:**

\*

\*\*\*

\*\*\*\*\*

* %s → placeholder for strings
* %n → moves to next line

## **2.24 Largest and Smallest Integers (Read five integers)**

import java.util.Scanner;

public class LargestSmallest {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int[] numbers = new int[5];

for (int i = 0; i < 5; i++) {

System.out.print("Enter integer " + (i + 1) + ": ");

numbers[i] = input.nextInt();

}

int largest = numbers[0];

int smallest = numbers[0];

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

if (numbers[i] > largest) largest = numbers[i];

if (numbers[i] < smallest) smallest = numbers[i];

}

System.out.printf("Largest: %d%n", largest);

System.out.printf("Smallest: %d%n", smallest);

}

}

## **2.25 Odd or Even**

import java.util.Scanner;

public class OddEven {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter an integer: ");

int number = input.nextInt();

if (number % 2 == 0) {

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

} else {

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

}

}

}

## **2.26 Multiples**

import java.util.Scanner;

public class Multiples {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the first integer: ");

int num1 = input.nextInt();

System.out.print("Enter the second integer: ");

int num2 = input.nextInt();

if (num1 % num2 == 0) {

System.out.println(num1 + " is a multiple of " + num2);

} else {

System.out.println(num1 + " is NOT a multiple of " + num2);

}

}

}

## **2.27 Checkerboard Pattern of Asterisks**

public class Checkerboard {

public static void main(String[] args) {

for (int row = 1; row <= 8; row++) {

for (int col = 1; col <= 8; col++) {

if ((row + col) % 2 == 0) {

System.out.print("\*");

} else {

System.out.print(" ");

}

}

System.out.println();

}

}

}

**2.28 Diameter, Circumference, and Area of a Circle**

import java.util.Scanner;

public class CircleCalculations {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the radius of the circle: ");

int r = input.nextInt();

System.out.printf("Diameter: %d%n", 2 \* r);

System.out.printf("Circumference: %f%n", 2 \* 3.14159 \* r);

System.out.printf("Area: %f%n", 3.14159 \* r \* r);

}

}

✅ Uses 3.14159 for π.

**2.29 Integer Value of a Character**

public class CharacterValues {

public static void main(String[] args) {

char[] chars = {'A','B','C','a','b','c','0','1','2','$','\*','+','/',' '};

for (char ch : chars) {

System.out.printf("The character %c has the value %d%n", ch, (int) ch);

}

}

}

✅ Displays **integer values (Unicode/ASCII)** for letters, digits, symbols, and space.

**2.30 Separating the Digits in a Five-Digit Integer**

import java.util.Scanner;

public class SeparateDigits {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter a five-digit integer: ");

int num = input.nextInt();

int digit1 = num / 10000;

int digit2 = (num / 1000) % 10;

int digit3 = (num / 100) % 10;

int digit4 = (num / 10) % 10;

int digit5 = num % 10;

System.out.printf("%d %d %d %d %d%n", digit1, digit2, digit3, digit4, digit5);

}

}

✅ Assumes the number has exactly five digits.

**2.31 Table of Squares and Cubes (0 to 10)**

public class SquaresCubes {

public static void main(String[] args) {

System.out.println("Number Square Cube");

for (int i = 0; i <= 10; i++) {

System.out.printf("%6d %7d %5d%n", i, i\*i, i\*i\*i);

}

}

}

✅ Produces a formatted table of **numbers, squares, and cubes**.

**2.32 Negative, Positive, and Zero Values**

import java.util.Scanner;

public class NumberCounts {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int positive = 0, negative = 0, zero = 0;

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

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

int num = input.nextInt();

if (num > 0) positive++;

else if (num < 0) negative++;

else zero++;

}

System.out.printf("Positive numbers: %d%n", positive);

System.out.printf("Negative numbers: %d%n", negative);

System.out.printf("Zeros: %d%n", zero);

}

}

✅ Counts **how many numbers are positive, negative, or zero** among five inputs.

**2.33 Body Mass Index (BMI) Calculator**

import java.util.Scanner;

public class BMICalculator {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter weight in pounds: ");

double weight = input.nextDouble();

System.out.print("Enter height in inches: ");

double height = input.nextDouble();

// BMI formula for pounds and inches: (weight / (height^2)) \* 703

double bmi = (weight / (height \* height)) \* 703;

System.out.printf("Your BMI is: %.2f%n", bmi);

System.out.println("BMI Categories:");

System.out.println("Underweight = <18.5");

System.out.println("Normal weight = 18.5–24.9");

System.out.println("Overweight = 25–29.9");

System.out.println("Obese = 30 or greater");

}

}

✅ Uses **pounds and inches**. Can also convert to kilograms/meters if preferred.

**2.34 World Population Growth Calculator**

import java.util.Scanner;

public class WorldPopulation {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter current world population: ");

double population = input.nextDouble();

System.out.print("Enter annual growth rate (in %): ");

double growthRate = input.nextDouble() / 100;

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

population = population \* (1 + growthRate);

System.out.printf("Estimated population after year %d: %.0f%n", year, population);

}

}

}

✅ Uses **compound growth formula**: Population = Population \* (1 + rate)

**2.35 Car-Pool Savings Calculator**

import java.util.Scanner;

public class CarPoolSavings {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter total miles driven per day: ");

double miles = input.nextDouble();

System.out.print("Enter cost per gallon of gasoline: ");

double costPerGallon = input.nextDouble();

System.out.print("Enter average miles per gallon: ");

double mpg = input.nextDouble();

System.out.print("Enter parking fees per day: ");

double parking = input.nextDouble();

System.out.print("Enter tolls per day: ");

double tolls = input.nextDouble();

double dailyCost = (miles / mpg) \* costPerGallon + parking + tolls;

System.out.printf("Your daily driving cost is: $%.2f%n", dailyCost);

}

}

✅ Calculates the **daily cost of driving** including gas, parking, and tolls.