**2.1 Fill in the Blanks**

a) A(n) **left brace {** begins the body of every method, and a(n) **right 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) **Space**, **tab** and **newline** are called white space.  
e) **Keywords** are reserved for use by Java.  
f) Java applications begin execution at method **main**.  
g) Methods **System.out.print**, **System.out.println** and **System.out.printf** display information in a command window.

**2.2 True or False**

a) *Comments cause the computer to print the text after the // on the screen when the program executes.*  
 **False.** Comments are ignored by the compiler and are not executed.

b) *All variables must be given a type when they’re declared.*  
 **True.** In Java, every variable must have a declared type.

c) *Java considers the variables number and NuMbEr to be identical.*  
 **False.** Java is case-sensitive; these are two different identifiers.

d) *The remainder operator (%) can be used only with integer operands.*  
 **False.** In Java, the % operator works with both integers and floating-point values.

e) *The arithmetic operators , /, %, + and - all have the same level of precedence.****False.*** *Multiplication (*), division (/), and remainder (%) have higher precedence than addition (+) and subtraction (-).

### ****2.3 Java Statements****

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

java

CopyEdit

int c, thisIsAVariable, q76354, number;

b) **Prompt the user** to enter an integer:

java

CopyEdit

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

c) **Input an integer** and assign it to variable value (assuming a Scanner variable named input exists):

java

CopyEdit

int value = input.nextInt();

d) **Print** "This is a Java program" using System.out.println:

java

CopyEdit

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

e) **Print** "This is a Java program" on two lines using System.out.printf with two %s specifiers:

java

CopyEdit

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

f) **Conditional display** if number is not equal to 7:

java

CopyEdit

if (number != 7) {

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

}

### ****2.4 Error Correction****

a) **Original:**

java

CopyEdit

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

**Error:** The semicolon after the if condition ends the if-statement prematurely.  
**Correction:**

java

CopyEdit

if (c < 7) {

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

}

b) **Original:**

java

CopyEdit

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

**Error:** The relational operator is incorrect; it should be >= instead of =>.  
**Correction:**

java

CopyEdit

if (c >= 7) {

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

}

### ****2.5 Declarations and Statements****

a) **Comment stating program purpose:**

java

CopyEdit

// This program calculates the product of three integers.

b) **Create a Scanner for input:**

java

CopyEdit

Scanner input = new Scanner(System.in);

c) **Declare variables:**

java

CopyEdit

int x, y, z, result;

d) **Prompt for the first integer:**

java

CopyEdit

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

e) **Read the first integer into x:**

java

CopyEdit

x = input.nextInt();

f) **Prompt for the second integer:**

java

CopyEdit

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

g) **Read the second integer into y:**

java

CopyEdit

y = input.nextInt();

h) **Prompt for the third integer:**

java

CopyEdit

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

i) **Read the third integer into z:**

java

CopyEdit

z = input.nextInt();

j) **Compute the product:**

java

CopyEdit

result = x \* y \* z;

k) **Display the product using System.out.printf:**

java

CopyEdit

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

### ****2.6 Complete Program****

Below is a complete Java program that calculates and prints the product of three integers:

java

CopyEdit

import java.util.Scanner;

public class ProductCalculator {

public static void main(String[] args) {

// This program calculates the product of three integers.

Scanner input = new Scanner(System.in);

int x, y, z, result;

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

x = input.nextInt();

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

y = input.nextInt();

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

z = input.nextInt();

result = x \* y \* z;

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

}

}

### ****2.7 Fill in the Blanks****

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 **expression** statements.  
d) The arithmetic operators with the same precedence as multiplication are **division** and **remainder**.  
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 Java Statements****

a) **Display a prompt on the same line:**

java

CopyEdit

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

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

java

CopyEdit

a = b \* c;

c) **Comment for a sample payroll calculation:**

java

CopyEdit

// This program performs a sample payroll calculation.

### ****2.9 True/False and Explanation****

a) Java operators are evaluated from left to right.  
 **True.** In Java, operands are evaluated from left to right (though operator precedence governs how the overall expression is computed).

b) The following are all valid variable names: under\_bar, m928134, t5, j7, her\_sales$, his\_$account\_total, a, b$, c, z and z2.  
 **True.** All these names follow Java's naming rules (starting with a letter, underscore, or dollar sign, and containing letters, digits, underscores, or dollar signs).

c) A valid Java arithmetic expression with no parentheses is evaluated from left to right.  
 **False.** Operator precedence and associativity rules determine the order of evaluation, not simply left-to-right.

d) The following are all invalid variable names: 3g, 87, 67h2, h22 and 2h.  
 **False.** While 3g, 87, 67h2, and 2h are invalid because they begin with a digit, **h22** is valid because it starts with a letter.

### ****2.10 Output Statements (Given x = 2 and y = 3)****

a)

java

CopyEdit

System.out.printf("x = %d%n", x);

**Output:** x = 2 (followed by a newline)

b)

java

CopyEdit

System.out.printf("Value of %d + %d is %d%n", x, x, (x + x));

**Output:** Value of 2 + 2 is 4 (followed by a newline)

c)

java

CopyEdit

System.out.printf("x =");

**Output:** x = (cursor remains on the same line)

d)

java

CopyEdit

System.out.printf("%d = %d%n", (x + y), (y + x));

Since addition is commutative, both (x+y) and (y+x) are 5.  
**Output:** 5 = 5 (followed by a newline)

### ****2.11 Identifying Modified Variables****

a) p = i + j + k + 7;  
 **Modified Variable:** p is assigned a new value.  
b) System.out.println("variables whose values are modified");  
 **No variables** are modified; this prints a string literal.  
c) System.out.println("a = 5");  
 **No variables** are modified; this prints a string literal.  
d) value = input.nextInt();  
 **Modified Variable:** value is assigned the input read from the user.

### ****2.12 Correct Java Statements for y = a·x³ + 7****

Given the equation:  
  y = axxx + 7\*  
The correct Java statements are:

a)

java

CopyEdit

y = a \* x \* x \* x + 7;

d)

java

CopyEdit

y = (a \* x) \* x \* x + 7;

e)

java

CopyEdit

y = a \* (x \* x \* x) + 7;

(The other options change the intended grouping and result.)

### ****2.13 Order of Evaluation****

a)

java

CopyEdit

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

**Evaluation Order:**

1. 3 \* 6 → 18
2. 18 / 2 → 9
3. 7 + 9 → 16
4. 16 - 1 → 15  
   **Final Value:** x = 15

b)

java

CopyEdit

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

**Evaluation Order:**

1. 2 % 2 → 0
2. 2 \* 2 → 4
3. 2 / 2 → 1
4. 0 + 4 → 4
5. 4 - 1 → 3  
   **Final Value:** x = 3

c)

java

CopyEdit

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

**Evaluation Order:**

1. Innermost: 9 \* 3 → 27
2. Then: 27 / 3 → 9
3. Next: 3 + 9 → 12
4. Then: 3 \* 9 → 27
5. Finally: 27 \* 12 → 324  
   **Final Value:** x = 324

### ****2.14 Displaying Numbers 1 to 4 on the Same Line****

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

java

CopyEdit

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

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

java

CopyEdit

System.out.print("1 ");

System.out.print("2 ");

System.out.print("3 ");

System.out.print("4");

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

java

CopyEdit

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

**2.15 Arithmetic**

A program that prompts for two integers and computes their sum, product, difference, and quotient (using integer division):

java

CopyEdit

import java.util.Scanner;

public class ArithmeticOperations {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int number1 = input.nextInt();

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

int number2 = input.nextInt();

int sum = number1 + number2;

int product = number1 \* number2;

int difference = number1 - number2;

int quotient = number1 / number2;

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);

input.close();

}

}

**2.16 Comparing Integers**

A program that compares two integers and displays the larger one (or a message if they are equal):

java

CopyEdit

import java.util.Scanner;

public class ComparingIntegers {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int number1 = input.nextInt();

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

int number2 = input.nextInt();

if (number1 > number2) {

System.out.printf("%d is larger%n", number1);

} else if (number2 > number1) {

System.out.printf("%d is larger%n", number2);

} else {

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

}

input.close();

}

}

**2.17 Arithmetic, Smallest and Largest**

A program that accepts three integers and then prints their sum, integer average, product, smallest, and largest value:

java

CopyEdit

import java.util.Scanner;

public class ThreeNumberStats {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int number1 = input.nextInt();

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

int number2 = input.nextInt();

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

int number3 = input.nextInt();

int sum = number1 + number2 + number3;

int average = sum / 3;

int product = number1 \* number2 \* number3;

// Find the smallest

int smallest = number1;

if (number2 < smallest) {

smallest = number2;

}

if (number3 < smallest) {

smallest = number3;

}

// Find the largest

int largest = number1;

if (number2 > largest) {

largest = number2;

}

if (number3 > largest) {

largest = number3;

}

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);

input.close();

}

}

**2.18 Displaying Shapes with Asterisks**

A program that prints a box, an oval, an arrow, and a diamond using asterisks:

java

CopyEdit

public class AsteriskShapes {

public static void main(String[] args) {

// Box

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

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

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

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

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

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

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

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

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

System.out.println(); // Blank line between shapes

// Oval

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

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

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

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

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

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

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

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

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

System.out.println(); // Blank line between shapes

// Arrow

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

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

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

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

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

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

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

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

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

System.out.println(); // Blank line between shapes

// Diamond

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

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

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

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

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

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

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

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

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

}

}

**2.19 through 2.23 Output Questions**

* **2.19:**

java

CopyEdit

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

**Prints:**

markdown

CopyEdit

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

* **2.20:**

java

CopyEdit

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

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

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

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

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

**Prints:**

markdown

CopyEdit

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*

* **2.21:**

java

CopyEdit

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

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

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

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

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

**Prints:** a single line with 15 asterisks combined, then a newline.

* **2.22:**

java

CopyEdit

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

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

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

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

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

**Prints:**

markdown

CopyEdit

\*\*\*\*

\*\*\*\*\*

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

*(The first line is the combination of "*" and "\*\**", and so on.)*

* **2.23:**

java

CopyEdit

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

**Prints:**

markdown

CopyEdit

\*

\*\*\*

\*\*\*\*\*

**2.24 Largest and Smallest Integers**

A program that reads five integers and finds the largest and smallest:

java

CopyEdit

import java.util.Scanner;

public class LargestAndSmallest {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int number1 = input.nextInt();

int largest = number1;

int smallest = number1;

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

int number2 = input.nextInt();

if (number2 > largest) largest = number2;

if (number2 < smallest) smallest = number2;

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

int number3 = input.nextInt();

if (number3 > largest) largest = number3;

if (number3 < smallest) smallest = number3;

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

int number4 = input.nextInt();

if (number4 > largest) largest = number4;

if (number4 < smallest) smallest = number4;

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

int number5 = input.nextInt();

if (number5 > largest) largest = number5;

if (number5 < smallest) smallest = number5;

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

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

input.close();

}

}

**2.25 Odd or Even**

A program that determines if an integer is odd or even:

java

CopyEdit

import java.util.Scanner;

public class OddOrEven {

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.printf("%d is even%n", number);

} else {

System.out.printf("%d is odd%n", number);

}

input.close();

}

}

**2.26 Multiples**

A program that checks if one number is a multiple of another:

java

CopyEdit

import java.util.Scanner;

public class Multiples {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int first = input.nextInt();

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

int second = input.nextInt();

if (first % second == 0) {

System.out.printf("%d is a multiple of %d%n", first, second);

} else {

System.out.printf("%d is not a multiple of %d%n", first, second);

}

input.close();

}

}

**2.27 Checkerboard Pattern of Asterisks**

A program that displays an 8-row checkerboard pattern:

java

CopyEdit

public class CheckerboardPattern {

public static void main(String[] args) {

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

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

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

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

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

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

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

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

}

}

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

A program that computes and displays circle measurements given an integer radius:

java

CopyEdit

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 (as an integer): ");

int radius = input.nextInt();

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

System.out.printf("Circumference: %f%n", 2 \* Math.PI \* radius);

System.out.printf("Area: %f%n", Math.PI \* radius \* radius);

input.close();

}

}

**2.29 Integer Value of a Character**

A program that prints the integer (Unicode) values for several characters:

java

CopyEdit

public class CharacterValues {

public static void main(String[] args) {

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

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

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

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

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

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

System.out.printf("The character %c has the value %d%n", '0', (int) '0');

System.out.printf("The character %c has the value %d%n", '1', (int) '1');

System.out.printf("The character %c has the value %d%n", '2', (int) '2');

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

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

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

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

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

}

}

**2.30 Separating the Digits in an Integer**

A program that breaks a five-digit number into individual digits and prints them with three spaces in between:

java

CopyEdit

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 number = input.nextInt();

int digit1 = number / 10000;

int digit2 = (number % 10000) / 1000;

int digit3 = (number % 1000) / 100;

int digit4 = (number % 100) / 10;

int digit5 = number % 10;

System.out.printf("%d %d %d %d %d%n",

digit1, digit2, digit3, digit4, digit5);

input.close();

}

}

**2.31 Table of Squares and Cubes**

A program that prints a table of numbers (0–10) along with their squares and cubes:

java

CopyEdit

public class SquaresAndCubes {

public static void main(String[] args) {

System.out.println("number\tsquare\tcube");

System.out.printf("%d\t%d\t%d%n", 0, 0\*0, 0\*0\*0);

System.out.printf("%d\t%d\t%d%n", 1, 1\*1, 1\*1\*1);

System.out.printf("%d\t%d\t%d%n", 2, 2\*2, 2\*2\*2);

System.out.printf("%d\t%d\t%d%n", 3, 3\*3, 3\*3\*3);

System.out.printf("%d\t%d\t%d%n", 4, 4\*4, 4\*4\*4);

System.out.printf("%d\t%d\t%d%n", 5, 5\*5, 5\*5\*5);

System.out.printf("%d\t%d\t%d%n", 6, 6\*6, 6\*6\*6);

System.out.printf("%d\t%d\t%d%n", 7, 7\*7, 7\*7\*7);

System.out.printf("%d\t%d\t%d%n", 8, 8\*8, 8\*8\*8);

System.out.printf("%d\t%d\t%d%n", 9, 9\*9, 9\*9\*9);

System.out.printf("%d\t%d\t%d%n", 10, 10\*10, 10\*10\*10);

}

}

**2.32 Negative, Positive and Zero Values**

A program that reads five integers and counts how many are negative, positive, or zero:

java

CopyEdit

import java.util.Scanner;

public class CountValues {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int negativeCount = 0;

int positiveCount = 0;

int zeroCount = 0;

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

int number1 = input.nextInt();

if (number1 < 0) {

negativeCount++;

} else if (number1 > 0) {

positiveCount++;

} else {

zeroCount++;

}

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

int number2 = input.nextInt();

if (number2 < 0) {

negativeCount++;

} else if (number2 > 0) {

positiveCount++;

} else {

zeroCount++;

}

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

int number3 = input.nextInt();

if (number3 < 0) {

negativeCount++;

} else if (number3 > 0) {

positiveCount++;

} else {

zeroCount++;

}

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

int number4 = input.nextInt();

if (number4 < 0) {

negativeCount++;

} else if (number4 > 0) {

positiveCount++;

} else {

zeroCount++;

}

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

int number5 = input.nextInt();

if (number5 < 0) {

negativeCount++;

} else if (number5 > 0) {

positiveCount++;

} else {

zeroCount++;

}

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

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

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

input.close();

}

}

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

A BMI calculator that takes weight in pounds and height in inches, computes BMI using the imperial formula, and prints BMI values along with the categories:

java

CopyEdit

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: ");

int weightInPounds = input.nextInt();

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

int heightInInches = input.nextInt();

int bmi = (weightInPounds \* 703) / (heightInInches \* heightInInches);

System.out.printf("BMI: %d%n", bmi);

System.out.println("BMI VALUES");

System.out.println("Underweight: less than 18.5");

System.out.println("Normal: between 18.5 and 24.9");

System.out.println("Overweight: between 25 and 29.9");

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

input.close();

}

}

**2.34 World Population Growth Calculator**

A program that calculates the estimated world population for the next five years based on the current population and an annual growth rate:

java

CopyEdit

import java.util.Scanner;

public class PopulationGrowth {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

long currentPopulation = input.nextLong();

System.out.print("Enter annual world population growth rate (percentage): ");

double growthRate = input.nextDouble();

// Convert growth rate to decimal

double growthRateDecimal = growthRate / 100;

long year1Population = currentPopulation + (long)(currentPopulation \* growthRateDecimal);

long year2Population = year1Population + (long)(year1Population \* growthRateDecimal);

long year3Population = year2Population + (long)(year2Population \* growthRateDecimal);

long year4Population = year3Population + (long)(year3Population \* growthRateDecimal);

long year5Population = year4Population + (long)(year4Population \* growthRateDecimal);

System.out.printf("World population after 1 year: %d%n", year1Population);

System.out.printf("World population after 2 years: %d%n", year2Population);

System.out.printf("World population after 3 years: %d%n", year3Population);

System.out.printf("World population after 4 years: %d%n", year4Population);

System.out.printf("World population after 5 years: %d%n", year5Population);

input.close();

}

}

**2.35 Car-Pool Savings Calculator**

A program that calculates your daily driving cost based on miles per day, gasoline cost, fuel efficiency, parking fees, and tolls:

java

CopyEdit

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 totalMiles = input.nextDouble();

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

double costPerGallon = input.nextDouble();

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

double milesPerGallon = input.nextDouble();

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

double parkingFees = input.nextDouble();

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

double tolls = input.nextDouble();

// Calculate daily driving cost

double dailyCost = (totalMiles / milesPerGallon) \* costPerGallon + parkingFees + tolls;

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

input.close();

}

}