**Answers to Chapter 3 Questions**

**Exercise 1**

**3.10: Comparing if and while statements**

* **Similarities**: Both if and while statements execute a block of code based on a condition.
* **Differences**:
  + if is a **single-selection statement** that executes the block once if the condition is true.
  + while is a **repetition statement** that continues executing the block repeatedly while the condition remains true.

**3.11: Integer Division in Java**

* When dividing one integer by another in Java, the result is an **integer** (the fractional part is discarded).
* Example: 5 / 2 results in 2, not 2.5.
* **Solution**: Convert at least one operand to a floating-point type (double or float).
* double result = 5.0 / 2; // Output: 2.5

**3.12: Two Ways to Combine Control Statements**

1. **Sequential execution** – Statements are executed in order.
2. **Control structures nesting** – One control structure is placed inside another, like an if statement inside a while loop.

**3.13: Repetition for Sum Calculations**

* **First 100 positive integers**: Use a for loop (definite repetition).
* **Arbitrary number of positive integers**: Use a while loop with user input (indefinite repetition).
* // Sum of first 100 positive integers
* int sum = 0;
* for (int i = 1; i <= 100; i++) {
* sum += i;
* }
* // Sum of arbitrary numbers
* Scanner scanner = new Scanner(System.in);
* int num, total = 0;
* while ((num = scanner.nextInt()) > 0) {
* total += num;
* }

**3.14: Preincrement vs Postincrement**

* **Preincrement (++x)**: Increments x **before** using it in an expression.
* **Postincrement (x++)**: Uses x **first**, then increments it.

**3.15: Correcting Errors**

a) **Error**: Semicolon (;) after if, and mismatched quotes.

if (age >= 65)

System.out.println("Age is greater than or equal to 65");

else

System.out.println("Age is less than 65");

b) **Error**: total is not initialized.

int x = 1, total = 0;

while (x <= 10) {

total += x;

++x;

}

c) **Error**: Missing {} around loop body.

while (x <= 100) {

total += x;

++x;

}

d) **Error**: y keeps increasing, causing an infinite loop.

while (y > 0) {

System.out.println(y);

--y; // Fix: Decrement y instead of incrementing

}

**Exercise 2**

**3.17: Gas Mileage Calculator (Java Code)**

import java.util.Scanner;

public class GasMileage {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int miles, gallons;

double totalMiles = 0, totalGallons = 0;

while (true) {

System.out.print("Enter miles driven (-1 to quit): ");

miles = scanner.nextInt();

if (miles == -1) break;

System.out.print("Enter gallons used: ");

gallons = scanner.nextInt();

double mpg = (double) miles / gallons;

totalMiles += miles;

totalGallons += gallons;

System.out.printf("Miles per gallon for this trip: %.2f%n", mpg);

System.out.printf("Combined miles per gallon: %.2f%n", totalMiles / totalGallons);

}

}

}

**3.18: Credit Limit Calculator**

import java.util.Scanner;

public class CreditLimit {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int accountNumber = input.nextInt();

System.out.print("Enter beginning balance: ");

int balance = input.nextInt();

System.out.print("Enter total charges: ");

int charges = input.nextInt();

System.out.print("Enter total credits: ");

int credits = input.nextInt();

System.out.print("Enter credit limit: ");

int creditLimit = input.nextInt();

int newBalance = balance + charges - credits;

System.out.println("New balance: " + newBalance);

if (newBalance > creditLimit) {

System.out.println("Credit limit exceeded.");

}

}

}