**Chapter 3**

**Exercise 1**

**3.10 Comparison of if and while Statements**

* **Similarity:** Both if and while statements control the flow of execution based on a condition.
* **Difference:** The if statement executes a block of code once if the condition is true, while the while statement repeats execution as long as the condition remains true.

**3.11 Integer Division in Java**

* When dividing one integer by another, Java performs integer division, discarding the fractional part.
* Example: 5 / 2 results in 2, not 2.5.
* To avoid this, at least one operand should be a floating-point type (double or float).

java

CopyEdit

double result = (double) 5 / 2; // Produces 2.5

**3.12 Combining Control Statements**

Control statements can be combined in two ways:

1. **Sequentially** – One statement follows another.
2. **Nesting** – One statement is placed inside another, such as a loop within a loop.

**3.13 Type of Repetition for Summation**

* **Fixed repetition (for loop)** is suitable for summing the first 100 positive integers.
* **Variable repetition (while loop)** is appropriate for summing an arbitrary number of integers.

**3.14 Preincrement vs. Postincrement**

* **Preincrement (++x)**: Increments the variable before using its value.
* **Postincrement (x++)**: Uses the variable's current value first, then increments.

**3.15 Identifying Errors**

* **(a) Error:** The semicolon after if terminates the condition early. Also, the else block has an incorrect closing quote.

java

CopyEdit

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 uninitialized.

java

CopyEdit

int x = 1, total = 0;

while (x <= 10) {

total += x;

++x;

}

* **(c) Error:** ++x; should be inside the loop block.

java

CopyEdit

while (x <= 100) {

total += x;

++x;

}

* **(d) Error:** ++y; should decrement (--y;) to prevent an infinite loop.

java

CopyEdit

while (y > 0) {

System.out.println(y);

--y;

}

**Exercise 2**

**3.17 (Gas Mileage Calculator)**

Develop a Java program using a Scanner to input miles driven and gallons used, calculating miles per gallon (MPG) for each trip and the combined MPG.

java

CopyEdit

import java.util.Scanner;

public class GasMileage {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int miles, gallons;

double totalMiles = 0, totalGallons = 0;

while (true) {

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

miles = input.nextInt();

if (miles == -1) break;

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

gallons = input.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("Total miles per gallon so far: %.2f%n", totalMiles / totalGallons);

}

input.close();

}

}

**3.18 (Credit Limit Calculator)**

Write a program to input customer details and determine if they exceeded their credit limit.

java

CopyEdit

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.");

}

input.close();

}

}

**3.19 (Sales Commission Calculator)**

Calculate salesperson earnings based on sales.

java

CopyEdit

import java.util.Scanner;

public class SalesCommission {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

double totalSales = 0;

while (true) {

System.out.print("Enter item value (or -1 to quit): ");

double sale = input.nextDouble();

if (sale == -1) break;

totalSales += sale;

}

double earnings = 200 + (0.09 \* totalSales);

System.out.printf("Total earnings: $%.2f%n", earnings);

input.close();

}

}

**3.20 (Salary Calculator)**

Calculate employees' gross pay.

java

CopyEdit

import java.util.Scanner;

public class SalaryCalculator {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

for (int i = 1; i <= 3; i++) {

System.out.print("Enter hours worked for employee " + i + ": ");

int hours = input.nextInt();

System.out.print("Enter hourly rate: ");

double rate = input.nextDouble();

double pay = (hours <= 40) ? (hours \* rate) : (40 \* rate + (hours - 40) \* 1.5 \* rate);

System.out.printf("Gross pay for employee %d: $%.2f%n", i, pay);

}

input.close();

}

}

**3.21 (Find the Largest Number)**

Find the largest of 10 numbers.

java

CopyEdit

import java.util.Scanner;

public class LargestNumber {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int largest = Integer.MIN\_VALUE;

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

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

int num = input.nextInt();

if (num > largest) {

largest = num;

}

}

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

input.close();

}

}

**3.25 (Checkerboard Pattern)**

Display a checkerboard pattern.

java

CopyEdit

public class Checkerboard {

public static void main(String[] args) {

for (int i = 0; i < 8; i++) {

if (i % 2 == 1) System.out.print(" ");

for (int j = 0; j < 8; j++) {

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

}

System.out.println();

}

}

}