**Assignment Chapter 3:**

**3.10**

The `if` single-selection statement allows conditional execution of a single block of code based on a Boolean expression. The `while` repetition statement repeatedly executes a block of code as long as the Boolean expression evaluates to `true`. They are similar in that both evaluate a Boolean expression. They differ in usage: `if` is for one-time decisions, while `while` is for repetition.

**3.11**

When a Java program divides one integer by another, the result is truncated to produce an integer, and any fractional part of the calculation is discarded. To avoid losing the fractional part, you can cast one or both integers to double before performing the division.

For example:

double result = (double) numerator / denominator;

**3.12**

Control statements in Java can be combined in two ways:

1. Sequentially: Control statements are executed one after another in sequence. For example:

if (x > 0) {

System.out.println("Positive");

}

while (x > 0) {

x--;

}

1. Nested: Control statements are placed inside other control statements. For example:

if (x > 0) {

while (x > 0) {

x--;

System.out.println(x);

}

}

**3.13**

To calculate the sum of the first 100 positive integers, counter-controlled repetition (a for loop) would be appropriate.

For example:

int sum = 0;

for (int i = 1; i <= 100; i++) {

sum += i;

}

To calculate the sum of an arbitrary number of positive integers, sentinel-controlled repetition (a while loop) would work better. For example:

java

int sum = 0;

int number;

Scanner input = new Scanner(System.in);

System.out.println("Enter a positive number (or -1 to stop):");

while ((number = input.nextInt()) != -1) {

sum += number;

}

**3.14**

Pre incrementing a variable (++x) increments the value of the variable before it is used in an expression, while post incrementing (x++) increments the value of the variable after it is used in the expression.

For example:

* Pre increment:

int x = 5;

int y = ++x; // x becomes 6, then y is assigned the value of 6.

* Post increment:

int x = 5;

int y = x++; // y is assigned the value of 5, then x becomes 6.

**3.15.**

Here are the errors and corrections for each code snippet:

**a)** Errors:The semicolon (;) after the if condition ends the if statement prematurely.

The else statement has a mismatched closing parenthesis (")").

Corrected Code:

if (age >= 65) {

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

} else {

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

}

**b)** Errors:The variable total is not initialized before using it.

Corrected Code:

int x = 1, total = 0; // Initialize total

while (x <= 10) {

total += x;

++x;

}

**c)** Errors:Missing curly braces ({}) for the while loop, so only the first statement is part of the loop.

Corrected Code:

int x = 1; // Assuming x is declared and initialized

while (x <= 100) {

total += x;

++x;

}

**d)** Errors:The increment operator (++y) causes an infinite loop as y is increasing rather than decreasing in a loop that expects y > 0.

Corrected Code:

while (y > 0) {

System.out.println(y);

--y; // Decrement y}

NB: 3.16 - 3.21, All in JavaNotes.java.

**3.22**

public static void tabularOutput() {

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

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

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

}

}

**3.23.**

public static void findTwoLargestNumbers() {

Scanner input = new Scanner(System.in);

int largest = Integer.MIN\_VALUE, secondLargest = Integer.MIN\_VALUE;

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

System.out.printf("Enter number %d: ", i);

int number = input.nextInt();

if (number > largest) {

secondLargest = largest;

largest = number;

} else if (number > secondLargest) {

secondLargest = number;

}

}

System.out.printf("The largest number is: %d%n", largest);

System.out.printf("The second largest number is: %d%n", secondLargest);

}

**3.24**.

public static void validateUserInput() {

Scanner input = new Scanner(System.in);

int choice;

do {

System.out.print("Enter 1 or 2: ");

choice = input.nextInt();

if (choice != 1 && choice != 2) {

System.out.println("Invalid input! Please try again.");

}

} while (choice != 1 && choice != 2);

System.out.printf("You entered a valid choice: %d%n", choice);

}

**3.25**

public static void checkerboardPattern() {

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

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

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

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

} else {

System.out.print(" ");

}

}

System.out.println();

}

}