**Chapter 3**

Exercise 1

3.10) Similarities: Both use Boolean expressions.

Both control flow decision.

In both statements, execution may skip code.

3.11) If the result of the division is not a whole number, the fractional part (decimal portion) is discarded returning only the integer quotient.

3.12) a) Control statements can be combined through nesting. This involves placing one control statements inside another. For example, you might nest an if statement within a while loop to check for additional conditions during iterations.

b) Control statements can be combined using logical operators such as && (AND), || (OR), and ! (NOT).

3.13) a) **For loop**: a for loop can iterate from 1 to 100, accumulating the sum of these numbers, Alternatively, sum can be calculated directly using the formula: **S = (n(n+1))/2**

**b) While Loop**: a while loop can repeatedly prompt the user for input until a termination condition (e.g., entering zero or a negative number) is met. The entered values are accumulated into a sum.

3.14) Pre-incrementing is when the variable is incremented before its value is used in the expression. **While**

Post-incrementing is when the variable is used in the expression first, then it is incremented.

3.15) a)

if (age >= 65) {

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

}

Else {

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

}

b)

int x = 1, total;

while (x <= 10) {

total += x; ++x;

}

c)

while (x <= 100)

total += x; ++x; d)

while (y > 0) {

System.out.println(y);

++y;