Here are the answers to the exercises:

Exercise 1

a) i = 1; while (i <= 10); ++i; }

Error: The while loop has a semicolon at the end, which means it will run indefinitely.

Correction:

i = 1;

while (i <= 10) {

++i;

}

b) for (k = 0.1; k != 1.0; k += 0.1) System.out.println(k); }

Error: The loop condition k != 1.0 may not be met due to floating-point precision issues.

Correction:

for (k = 0.1; k <= 1.0; k += 0.1) {

System.out.println(k);

}

c) switch (n) { case 1: System.out.println("The number is 1"); case 2: System.out.println("The number is 2"); break; default: System.out.println("The number is not 1 or 2"); break; } }

Error: The case 1 block is missing a break statement, which means it will fall through to the case 2 block.

Correction:

switch (n) {

case 1:

System.out.println("The number is 1");

break;

case 2:

System.out.println("The number is 2");

break;

default:

System.out.println("The number is not 1 or 2");

break;

}

d) The following code should print the values 1 to 10: n = 1; while (n < 10) System.out.println(n++); }

Error: The loop condition n < 10 will only print values up to 9.

Correction:

n = 1;

while (n <= 10) {

System.out.println(n++);

}

Exercise 2

4.5 The four basic elements of counter-controlled repetition are:

1. Initialization: The counter is initialized to a starting value.

2. Condition: The loop condition is checked to determine whether the loop should continue.

3. Increment/Decrement: The counter is incremented or decremented after each iteration.

4. Body: The code inside the loop is executed.

4.6 Both while and for statements are used for repetition, but they differ in their syntax and usage:

- while statement: The loop condition is checked before each iteration, and the loop body is executed if the condition is true.

- for statement: The loop initialization, condition, and increment/decrement are combined into a single statement.

4.7 A do...while statement is more appropriate when the loop body needs to execute at least once, regardless of the loop condition.

For example, in a menu-driven program, you might want to display the menu and get user input at least once before checking the loop condition.

4.8 Both break and continue statements are used to control the flow of loops:

- break statement: Exits the loop entirely and transfers control to the next statement after the loop.

- continue statement: Skips the current iteration and transfers control to the next iteration of the loop.

4.9a) For (i = 100, i >= 1, i++) System.out.println(i); }

Error: The loop condition i >= 1 will never be false, causing an infinite loop.

Correction:

for (i = 100; i >= 1; i--) {

System.out.println(i);

}

b) Error: The case 0 block is missing a break statement, which means it will fall through to the case 1 block.

Correction:

switch (value % 2) {

case 0:

System.out.println("Even integer");

break;

case 1:

System.out.println("Odd integer");

break;

}

c) Error: The loop