**CHAPTER 4:**

**Exercise 1:**

1. **a)**

**java**

**i = 1;**

**while (i <= 10);**

**++i;**

**}**

* + **Error: The semicolon (;) after while (i <= 10) prematurely ends the loop, and the closing brace } is misplaced.**
  + **Correction: Remove the semicolon and properly include the closing brace after the loop's body:**

**java**

**i = 1;**

**while (i <= 10) {**

**++i;**

**}**

1. **b)**

**java**

**for (k = 0.1; k != 1.0; k += 0.1)**

**System.out.println(k);**

* + **Error: Floating-point numbers like 0.1 are not precise, so comparing k != 1.0 might lead to an infinite loop.**
  + **Correction: Use a tolerance value for comparison:**

**java**

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

**System.out.println(k);**

1. **c)**

**java**

**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: Missing break after case 1, causing fall-through behavior.**
  + **Correction: Add break after case 1:**

**java**

**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;**

**}**

1. **d)**

**java**

**n = 1;**

**while (n < 10)**

**System.out.println(n++);**

* + **Error: The loop prints numbers from 1 to 9 instead of 1 to 10.**
  + **Correction: Adjust the condition to n <= 10:**

**java**

**n = 1;**

**while (n <= 10)**

**System.out.println(n++);**

**Exercise 2:**

1. **4.5: Four basic elements of counter-controlled repetition:**
   * **Initialization: Set the counter variable to its starting value.**
   * **Condition: Test whether the loop should continue.**
   * **Increment/Decrement: Adjust the counter variable after each iteration.**
   * **Body: The statements executed within the loop.**
2. **4.6: Comparison of while and for statements:**
   * **while: Ideal for situations where the number of repetitions isn't known beforehand. The condition is checked before executing the loop.**
   * **for: Suitable for counter-controlled loops where the number of repetitions is known. Combines initialization, condition, and increment/decrement in one line.**
3. **4.7: Situation for do...while:**
   * **Use do...while when you need to execute the loop at least once before checking the condition. For example, prompting a user to input a valid number:**

**java**

**int input;**

**do {**

**System.out.println("Enter a number greater than zero:");**

**input = scanner.nextInt();**

**} while (input <= 0);**

1. **4.8: Comparison of break and continue:**
   * **break: Exits the loop entirely and skips remaining iterations.**
   * **continue: Skips the current iteration and proceeds to the next one.**

**Correct errors in 4.9:**

1. **a)**

**java**

**For (i = 100, i >= 1, i++)**

**System.out.println(i);**

* + **Error: Incorrect capitalization of For and improper use of commas.**
  + **Correction:**

**java**

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

**System.out.println(i);**

1. **b)**

**java**

**switch (value % 2) {**

**case 0:**

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

**case 1:**

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

**}**

* + **Error: Missing break in both cases.**
  + **Correction:**

**java**

**switch (value % 2) {**

**case 0:**

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

**break;**

**case 1:**

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

**break;**

**}**

1. **c)**

**java**

**for (i = 19; i >= 1; i += 2)**

**System.out.println(i);**

* + **Error: Increment i instead of decrementing it.**
  + **Correction:**

**java**

**for (i = 19; i >= 1; i -= 2)**

**System.out.println(i);**

1. **d)**

**java**

**counter = 2;**

**do {**

**System.out.println(counter);**

**counter += 2;**

**} While (counter < 100);**

* + **Error: Misuse of capitalization in While.**
  + **Correction:**

**java**

**counter = 2;**

**do {**

**System.out.println(counter);**

**counter += 2;**

**} while (counter <= 100);**

**4.10: Explanation of Printing.java:**

* **This program prints a grid of 10 rows, with each row containing 5 @ symbols. It uses nested loops: the outer loop iterates over rows, and the inner loop handles the printing of @ symbols in each row.**