**Exercise 1: Error Identification and Correction**

1. **Code:**

java

CopyEdit

i = 1;

while (i <= 10);

++i;

}

**Errors:**

* + Semicolon after while (i <= 10); makes it an empty loop.
  + There's an unmatched closing brace }. **Correction:**

java

CopyEdit

int i = 1;

while (i <= 10) {

++i;

}

1. **Code:**

java

CopyEdit

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

System.out.println(k);

**Error:**

* + Floating-point comparison k != 1.0 is unreliable due to precision issues. **Correction:**

java

CopyEdit

for (double k = 0.1; k < 1.0 + 1e-9; k += 0.1)

System.out.println(k);

1. **Code:**

java

CopyEdit

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. **Correction:**

java

CopyEdit

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. **Code:**

java

CopyEdit

n = 1;

while (n < 10)

System.out.println(n++);

**Error:**

* + Off-by-one error: prints 1 to 9, not 1 to 10. **Correction:**

java

CopyEdit

int n = 1;

while (n <= 10)

System.out.println(n++);

**Exercise 2 Answers**

**4.5**: Four basic elements of counter-controlled repetition:

1. **Control variable** initialization.
2. **Loop-continuation condition**.
3. **Update** of the control variable.
4. **Body** of the loop.

**4.6**:

* while: Use when the number of iterations isn't known in advance.
* for: Use when the number of iterations is known or count-controlled.

**4.7**: Use do...while when the loop body must execute **at least once** (e.g., menu-driven input).

**4.8**:

* break: Exits the loop immediately.
* continue: Skips current iteration and continues with the next one.

**4.9**:

a)

java

CopyEdit

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

System.out.println(i);

b)

java

CopyEdit

switch (value % 2) {

case 0:

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

break;

case 1:

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

break;

}

c)

java

CopyEdit

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

System.out.println(i);

d)

java

CopyEdit

int counter = 2;

do {

System.out.println(counter);

counter += 2;

} while (counter <= 100);

**4.10**: The program prints a **10x5 grid** of the @ character.

**4.11**: A program that takes n inputs and finds the smallest. Needs a loop with comparison logic.

**4.12**:

java

CopyEdit

int product = 1;

for (int i = 1; i <= 15; i += 2)

product \*= i;

System.out.println("Product: " + product);

**4.13**:

java

CopyEdit

long factorial = 1;

for (int i = 1; i <= 20; i++) {

factorial \*= i;

System.out.println(i + "! = " + factorial);

}

// Factorial of 100 may cause overflow, even with `long`.

**4.14**: Modify compound interest to loop through interest rates 5–10%.