**Snippet 1:**

public class InfiniteForLoop {

public static void main(String[] args) {

for (int i = 0; i < 10; i--) {

System.out.println(i);

}

}

}

**// Error to investigate: Why does this loop run infinitely? How should the loop control variable be adjusted?1.Error**

**2.explanation**

1. The loop runs infinitely because because the of the condition(i<10) .

2. It can stop by increamenting loop control variable (++)

**3.Fix**

public class InfiniteForLoop {

public static void main(String[] args) {

for (int i = 0; i < 10; i++) {

System.out.println(i);

}

}

}

**Snippet 2:**

public class IncorrectWhileCondition {

public static void main(String[] args) {

int count = 5;

while (count = 0) {

System.out.println(count);

count--;

}

}

}

**// Error to investigate: Why does the loop not execute as expected? What is the issue with the condition in the**

**`while` loop?**

**2.explanation**

Because of the incompatible types int cannot be converted to boolean

**3.Fix**

public class IncorrectWhileCondition {

public static void main(String[] args) {

int count = 5;

while (count != 0) {

System.out.println(count);

count--;

}

}

}

**Snippet 3 :**

public class DoWhileIncorrectCondition {

public static void main(String[] args) {

int num = 0;

do {

System.out.println(num);

num++;

} while (num > 0);

}

}

**// Error to investigate: Why does the loop only execute once? What is wrong with the loop condition in the `do**

**2.explanation**

Because in do while loop, the loop will always executes once even if the the condition is false

**3.Fix**

public class DoWhileIncorrectCondition {

public static void main(String[] args) {

int num = 0;

do {

System.out.println(num);

num++;

} while (num < 5);

}

}

**Snippet 4:**

public class OffByOneErrorForLoop {

public static void main(String[] args) {

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

System.out.println(i);

}

// Expected: 10 iterations with numbers 1 to 10

// Actual: Prints numbers 1 to 10, but the task expected only 1 to 9

}

}

**// Error to investigate: What is the issue with the loop boundaries? How should the loop be adjusted to meet the**

**expected output?**

**Explanation :**

1. The loop starts from 1 to 10 becaue of <= operator.

**Fix:**

public class OffByOneErrorForLoop {

public static void main(String[] args) {

for (int i = 1; i <= 9; i++) {

System.out.println(i);

}

}

}

**Snippet 5:**

public class WrongInitializationForLoop {

public static void main(String[] args) {

for (int i = 10; i >= 0; i++) {

System.out.println(i);

}

}

}

**// Error to investigate: Why does this loop not print numbers in the expected order? What is the problem with the**

**initialization and update statements in the `for` loop?**

**Explantaion:**

**.** The loop runs infinitely because because the of the condition(i=>0).i will get incrementing due to increment operator

**Fix :**

public class WrongInitializationForLoop {

public static void main(String[] args) {

for (int i = 0; i <= 10; i++) {

System.out.println(i);

}

}

}

**Snippet 6:**

public class MisplacedForLoopBody {

public static void main(String[] args) {

for (int i = 0; i < 5; i++)

System.out.println(i);

System.out.println("Done");

}

}

**// Error to investigate: Why does "Done" print only once, outside the loop? How should the loop body be enclosed to**

**include all statements within the loop?**

**Explanation:**

The for loop is only controlling the first statement (System.out.println(i);).

IT means "Done" is printed only once after the loop completes.

**Fix:**

public class MisplacedForLoopBody {

public static void main(String[] args) {

for (int i = 0; i < 5; i++) {

System.out.println(i);

System.out.println("Done");

}

}

}

**Snippet 7:**

public class UninitializedWhileLoop {

public static void main(String[] args) {

int count;

while (count < 10) {

System.out.println(count);

count++;

}

}

}

**// Error to investigate: Why does this code produce a compilation error? What needs to be done to initialize the loop**

**variable properly?**

**Explantion:**

In java, we have to initialize variable in or to use it

**Fix :**

public class UninitializedWhileLoop {

public static void main(String[] args) {

int count = 0;

while (count < 10) {

System.out.println(count);

count++;

}

}

}

**Snippet 8:**

public class OffByOneDoWhileLoop {

public static void main(String[] args) {

int num = 1;

do {

System.out.println(num);

num--;

} while (num > 0);

}

}

**// Error to investigate: Why does this loop print unexpected numbers? What adjustments are needed to print the**

**numbers from 1 to 5?**

**Explanation: to print numbers from 1 to 5 , initialization should be from 1, condition will be <=5,and operator hould be i++**

**Fix:**

public class OffByOneDoWhileLoop {

public static void main(String[] args) {

int num = 1;

do {

System.out.println(num);

Num++;

} while (num <=5);

}

}

**Snippet 9:**

public class InfiniteForLoopUpdate {

public static void main(String[] args) {

for (int i = 0; i < 5; i += 2) {

System.out.println(i);

}

}

}

**// Error to investigate: Why does the loop print unexpected results or run infinitely? How should the loop update**

**expression be corrected?**

**Explanation:**

The code gives an o/p as 0,2,4 because we are incrementing by 2 not 1

**Fix:**

public class InfiniteForLoopUpdate {

public static void main(String[] args) {

for (int i = 0; i < 5; i++) {

System.out.println(i);

}

}

}

**Snippet 10:**

public class IncorrectWhileLoopControl {

public static void main(String[] args) {

int num = 10;

while (num = 10) {

System.out.println(num);

num--;

}

}

}

**// Error to investigate: Why does the loop execute indefinitely? What is wrong with the loop condition?**

**Explanation:**

in while loop, while must evaluate a condition in order to execute it

**Fix:**

public class IncorrectWhileLoopControl {

public static void main(String[] args) {

int num = 10;

while (num == 10) {

System.out.println(num);

num--;

}

}

}

**Snippet 11:**

public class IncorrectLoopUpdate {

public static void main(String[] args) {

int i = 0;

while (i < 5) {

System.out.println(i);

i += 2; // Error: This may cause unexpected results in output

}

}

}

**// Error to investigate: What will be the output of this loop? How should the loop variable be updated to achieve the**

**desired result?**

**Error:**

Output will be 0,2,4

**Fix:**

public class CorrectLoopUpdate {

public static void main(String[] args) {

int i = 0;

while (i < 5) {

System.out.println(i);

i++;

}

}

}

**Snippet 12:**

public class LoopVariableScope {

public static void main(String[] args) {

for (int i = 0; i < 5; i++) {

int x = i \* 2;

}

System.out.println(x); // Error: 'x' is not accessible here

}

}

**// Error to investigate: Why does the variable 'x' cause a compilation error?**

**Explanation:**

Because the variable is declared in the loop itself. We cannot access it outside loop

**Fix:**

public class LoopVariableScope {

public static void main(String[] args) {

int x = 0; // Declare x outside the loop

for (int i = 0; i < 5; i++) {

x = i \* 2;

}

System.out.println(x);

}

}