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 . The loop went into an infinite loop because the condition was incorrect.

Snippet 1:

public class InfiniteForLoop {

public static void main(String[] args) {

for (int i = 10; i >=5; i--) {

System.out.println(i);

}

}

}

Output:- 10

10

9

8

7

6

5

**Snippet 2:**

public class IncorrectWhileCondition {

public static void main(String[] args) {

int count = 5;

while (count = 0) { // incompatibale type error

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?

Eroor

Main.java:5: error: incompatible types: int cannot be converted to boolean

// write code

public class Main {

public static void m IncorrectWhileCondition(String[] args) {

int count = 5;

while (count > 0) {

System.out.println(count);s

count--;

}

}

}

**Snippet 3:**

public class DoWhileIncorrectCondition {

public static void main(String[] args) {

int num = 0;

do {

System.out.println(num);

num++;

} while (num > 0); //infinite loop

}

}

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

**while` loop?**

**The loop is infinite because the condition is true and there is no endpoint to stop it**

**Write code:-**

public class DoWhileIncorrectCondition {

public static void main(String[] args) {

int num = 0;

do {

System.out.println(num);

num++;

} while (num <= 6); //infinite loop

}

}

**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?

**Solution 1 to 9 is :-**

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

}

}

**Output :**

**Snippet 5:**

**public class WrongInitializationForLoop {**

**public static void main(String[] args) {**

**for (int i = 10; i >= 0; i++) { // infinite loop**

**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?**

**Write code :-**

**public class WrongInitializationForLoop {**

**public static void main(String[] args) {**

**for (int i = 10; I <= 15; i++) {**

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

**}**

**}**

**}**

**Output :- 10**

**11**

**12**

**13**

**14**

**15**

**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?**

**Code**

public class MisplacedForLoopBody {

public static void main(String[] args) {

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

System.out.println(i);

System.out.println("Done");

}

}

}

**Output :-**

**0**

**Done**

**1**

**Done**

**2**

**Done**

**3**

**Done**

**4**

**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?**

**Main.java:4: error: variable count might not have been initialized**

**while (count < 10) {**

**code:-**

public class UninitializedWhileLoop {

public static void main(String[] args) {

int count;

count = 5;

while (count < 10) {

System.out.println(count);

count++;

}

}

}

**Output :- 5**

**6**

**7**

**8**

**9**

**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?**

**Code :-**

public class OffByOneDoWhileLoop {

public static void main(String[] args) {

int num = 1;

do {

System.out.println(num);

**Num++;**

} while (**num <= 5**);

}

}

**Output :-**

**1**

**2**

**3**

**4**

**5**

**Snippet 9:**

public class InfiniteForLoopUpdate {

public static void main(String[] args) {

for (int i = 0; i < 5; i += 2) { // Increases i by 2 each time

System.out.println(i);

}

}

}

Output :- 0

2

4

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

expression be corrected?

**Code :-**

public class InfiniteForLoopUpdate {

public static void main(String[] args) {

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

System.out.println(i);

}

}

}

Output –

0

1

2

3

4

**Snippet 10:**

public class IncorrectWhileLoopControl {

public static void main(String[] args) {

int num = 10;

while (num = 10) { // incompatible data type

System.out.println(num);

num--;

}

}

}

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

**Error**

**Main.java:5: error: incompatible types: int cannot be converted to boolean**

**while (num = 10) {**

public class IncorrectWhileLoopControl {

public static void main(String[] args) {

int num = 10;

while (num == 10) { // == operator checks if two values are equal

System.out.println(num);

num--;

}

}

}

Output :- 10

**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

}

}

}

**Output = 0**

**2**

**4**

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

**desired result?**

**// code -**

public class IncorrectLoopUpdate {

public static void main(String[] args) {

int i = 0;

while (i < 5) {

System.out.println(i);

**i ++;**

}

}

}

Output = 0

1

2

3 4

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? How does scope

**Error :-**

**Main.java:7: error: cannot find symbol**

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

**^**

**symbol: variable x**

**location: class Main**

**code:-**

public class LoopVariableScope {

public static void main(String[] args) {

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

int x = i \* 2;

System.out.println(x); // write in inside loop

}

}

}

**Output :- 0**

**2**

**4**

**6**

**8**