**Snippet 1:**

public class InfiniteForLoop {

public static void main(String[] args) {

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

System.out.println(i);

}

}

}

Error output:

Infinite Loop

Solution:

The loop is running infinitely because we are decrementing i thus i is always smaller than 10. We need to increment it instead.

Modified Code:

public class InfiniteForLoop {

public static void main(String[] args) {

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

System.out.println(i);

}

}

}

Output:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>java InfiniteForLoop

0

1

2

3

4

5

6

7

8

9

**Snippet 2:**

public class IncorrectWhileCondition {

public static void main(String[] args) {

int count = 5;

while (count = 0) {

System.out.println(count);

count--;

}

}

}

Compile time error:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>javac IncorrectWhileCondition.java

IncorrectWhileCondition.java:4: error: incompatible types: int cannot be converted to boolean

while (count = 0) {

^

1 error

Solution:

There needs to be a Boolean value inside the while loop condition. We can use count>0 instead.

Modified code:

public class IncorrectWhileCondition {

public static void main(String[] args) {

int count = 5;

while (count > 0) {

System.out.println(count);

count--;

}

}

}

Output:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>java IncorrectWhileCondition

5

4

3

2

1

**Snippet 3:**

public class DoWhileIncorrectCondition {

public static void main(String[] args) {

int num = 0;

do {

System.out.println(num);

num++;

} while (num > 0);

}

}

Error output: Infinite loop

Solution: Instead of num > 0 which is always true, we can use num < 10 instead so that the condition will eventually become false and loop will be exited.

Modified code:

public class DoWhileIncorrectCondition {

public static void main(String[] args) {

int num = 0;

do {

System.out.println(num);

num++;

} while (num < 10);

}

}

Output:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>java DoWhileIncorrectCondition

0

1

2

3

4

5

6

7

8

9

**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

}

}

Solution: Currently printing 1 to 10. To print 1 to 9, we can use i < 10 instead.

Modified Code:

public class OffByOneErrorForLoop {

public static void main(String[] args) {

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

System.out.println(i);

}

}

}

Output:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>java OffByOneErrorForLoop

1

2

3

4

5

6

7

8

9

**Snippet 5:**

public class WrongInitializationForLoop {

public static void main(String[] args) {

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

System.out.println(i);

}

}

}

Error output: Infinite loop

Solution: We can decrement i instead so that the condition i >=0 will become false eventually.

Modified code:

public class WrongInitializationForLoop {

public static void main(String[] args) {

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

System.out.println(i);

}

}

}

Output:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>java WrongInitializationForLoop

10

9

8

7

6

5

4

3

2

1

0

**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");

}

}

Solution: Only the first statement after the for loop becomes part of it if the body is not enclosed in curly braces. To print Done 5 times as well, we need to include it in the body of the for loop.

Modified 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:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>java MisplacedForLoopBody

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++;

}

}

}

Compile time error:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>javac UninitializedWhileLoop.java

UninitializedWhileLoop.java:6: error: variable count might not have been initialized

while (count < 10) {

^

1 error

Solution: We need to initialize count with some value e.g. 0 before comparing with 10.

Modified code:

public class UninitializedWhileLoop {

public static void main(String[] args) {

int count=0;

while (count < 10) {

System.out.println(count);

count++;

}

}

}

Output:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>java UninitializedWhileLoop

0

1

2

3

4

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);

}

}

Solution: It is currently printing only 1. To print 1 to 5, we need to increment num instead of decrementing and use condition as num<=5.

Modified code:

public class OffByOneDoWhileLoop {

public static void main(String[] args) {

int num = 1;

do {

System.out.println(num);

num++;

} while (num <= 5);

}

}

Output:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>java OffByOneDoWhileLoop

1

2

3

4

5

**Snippet 9:**

public class InfiniteForLoopUpdate {

public static void main(String[] args) {

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

System.out.println(i);

}

}

}

Solution: It is currently printing 0 2 4 as we are incrementing i by 2. To print 0 to 4, we can increment by just 1 instead.

Modified code:

public class InfiniteForLoopUpdate {

public static void main(String[] args) {

for (int i = 0; i < 5; i += 1) {

System.out.println(i);

}

}

}

Output:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>java InfiniteForLoopUpdate

0

1

2

3

4

**Snippet 10:**

public class IncorrectWhileLoopControl {

public static void main(String[] args) {

int num = 10;

while (num = 10) {

System.out.println(num);

num--;

}

}

}

Solution: num = 10 is not valid as it is not a Boolean value. We can use num > 0 instead to print 10 to 1.

Modified code:

public class IncorrectWhileLoopControl {

public static void main(String[] args) {

int num = 10;

while (num > 0) {

System.out.println(num);

num--;

}

}

}

Output:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>java IncorrectWhileLoopControl

10

9

8

7

6

5

4

3

2

1

**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

}

}

}

Solution: It is printing 0 2 4 as we are incrementing i by 2. Instead we can increment it by 1.

Modified code:

public class IncorrectLoopUpdate {

public static void main(String[] args) {

int i = 0;

while (i < 5) {

System.out.println(i);

i += 1;

}

}

}

Output:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>java IncorrectLoopUpdate

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

}

}

Compile time error:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>javac LoopVariableScope.java

LoopVariableScope.java:7: error: variable x might not have been initialized

System.out.println(x);

^

1 error

Solution: As x is declared inside the for loop, its scope is limited to that loop only. To access it outside, we need to declare and initialize it outside as well.

Modified code:

public class LoopVariableScope {

public static void main(String[] args) {

int x=0;

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

x = i \* 2;

}

System.out.println(x);

}

}

Output:

C:\Users\Saurabh\Desktop\CDAC\Assignment 3>java LoopVariableScope

8