Assignment 3

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?

this loop run for infinite loop because loop control variable i is decreasing (i--) while the condition is i < 10. Since i is always decreasing, condition will never get terminated causing an infinite loop.

modify loop control variable i,e increment i instead of decreasing

corrected code:

public class InfiniteForLoop {

public static void main(String[] args) {

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

System.out.println(i);

}

}

}

O/p

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--;

}

}

}

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

`while` loop?

error: incompatible types: int cannot be converted to boolean

while (count = 0) {

^

assignment (=) operator is there in while loop instead of a comparison operators. In Java, count = 0 assigns 0 to count, which is not a valid boolean expression, leading to a compilation error.

Corrected code:

public class IncorrectWhileCondition {

public static void main(String[] args) {

int count = 5;

while (count > 0) {

System.out.println(count);

count--;

}

}

}

O/P : 5

4

3

2

1

-----------------------------------------------------------------------------------------------------------------------------

public class DoWhileIncorrectCondition {

public static void main(String[] args) {

int num = 0;

do {

System.out.println(num);

num++;

} while (num > 0);

}

}

it will run for infinite loop as num is being incremented so while condition will as it is num>0 never terminate causes infinite loop

Corrected Code :

public class DoWhileCorrected {

public static void main(String[] args) {

int num = 0;

do {

System.out.print(num + " ");

num++;

} while (num < 10); // Condition corrected to prevent infinite loop

}

}

O/p: 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.print(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?

Actual o/P: 1 2 3 4 5 6 7 8 9 10

Expected code with tas expected o/p

public class OffByOneErrorForLoop {

public static void main(String[] args) {

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

System.out.print(i + " ");

}

// Expected: 10 iterations with numbers 1 to 9

}

}

O/P :

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 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

This causes an infinite loop because i keeps increasing and never get terminated

Corrected Code:

public class WrongInitializationForLoop {

public static void main(String[] args) {

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

System.out.print(i + " ");

}

}

}

O/P

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");

}

}

// 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?

------------------------------------------------------------------------------------------------------------------

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?

compilation error: variable count might not have been initialized

while (count < 10) {

^

Corrected code :

public class UninitializedWhileLoop {

public static void main(String[] args) {

int count=0;

while (count < 10) {

System.out.print(count + " ");

count++;

}

}

}

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);

}

}

// Error to investigate: Why does this loop print unexpected numbers?

The loop prints only 1 then num -- and becomes 0 which cause to exit loop as it is do while loop, instead of counting up to 5.

What adjustments are needed to print the numbers from 1 to 5?

public class OffByOneDoWhileLoop {

public static void main(String[] args) {

int num = 1;

do {

System.out.println(num);

num++;

} while (num <= 5);

}

}

O/p:

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.print(i + " ");

}

}

}

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

expression be correcte

No any error it runs successfully

O/P : 0 2 4

=== Code Execution Successful ===

----------------------------------------------------------------------------------------------------------------------------------

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?

No error

O/P :

0

2

4

=== Code Execution Successful ===

-------------------------------------------------------------------------------------------------------------------------------------

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

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

^

symbol: variable x

location: class LoopVariableScope

1 error

local scope(declared inside for loop so scope upto only for loop) so it is not accessible outside

-------------------------------------------------------------------------------------------------------------------------------------