Assignment 3

22 Aug 2024

**Snippet 1 –**

public class InfiniteForLoop {

public static void main(String[] args) {

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

System.out.println(i);

}

}

}

**Why does this loop run infinitely?**

This loop is running infinitely because i always satisfies the condition.

**How should the loop control variable be adjusted?**

If we do i++ in place of i—the condition gets satisfy after 10 iterations and program get terminate.

**Modified Snippet 1 –**

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--;

} } }

**Why does the loop not execute as expected?**

Because in while() loop we need to give condition which leads to result in Boolean value. We need to use conditional operator in while loop.

**Modified Snippet 2-**

public class IncorrectWhileCondition {

public static void main(String[] args) {

int count = 5;

while (count > 0) { //here it will check the condition

System.out.println(count);

count--;

} } }

**Snippet 3 –**

public class InfiniteForLoop {

public static void main(String[] args) {

int num = 0;

do {

System.out.println(num);

num++;

} while (num < 0);

}

}

**Why does the loop only execute once?**

Because in while condition is false. In do {} we are doing num++ that implies in while condition gets false and program terminated.

**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

} }

**What is the issue with the loop boundaries? How should the loop be adjusted to meet the expected output?**

For getting expected output 1 to 9. We need to change the condition in for loop. i.e. for (int i = 1; i <10; i++) if we put this condition we get expected output.

**Modified Snippet 4-**

public class OffByOneErrorForLoop {

public static void main(String[] args) {

for (int i = 1; i < 10; i++) { //it prints the expected outputs

System.out.println(i);

}

// Expected: 10 iterations with numbers 1 to 10

**Snippet 5 –**

public class WrongInitializationForLoop {

public static void main(String[] args) {

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

System.out.println(i);

} } }

**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?**

It will print the values from 10 because we initializing the i with 10 and goes to infinite loop.

**Updated for loop-**

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

this will print values from t to 10.

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");

} }

**Why does "Done" print only once, outside the loop?**

“Done” is printed only once because it is outside of the loop.

**How should the loop body be enclosed to include all statements within the loop?**

To include all statements within the loop we have to use { }.

If there is more than one statement in for loop we have to use curly braces.

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++;

} } }

**Why does this code produce a compilation error?**

Not initialized the count value.

**What needs to be done to initialize the loop variable properly?**

To initialize value of count we need to assign any value to count variable which satisfy the condition.

**Snippet 8 –**

public class OffByOneDoWhileLoop {

public static void main(String[] args) {

int num = 1;

do { System.out.println(num);

num--;

} while (num > 0);

} }

**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 < 6);

} }

**Snippet 9 –**

public class InfiniteForLoopUpdate {

public static void main(String[] args) {

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

System.out.println(i);

} } }

**Output –** 0 ,2 , 4

**Snippet 10 –**

public class IncorrectWhileLoopControl {

public static void main(String[] args) {

int num = 10;

while (num = 10) { **//error- incompatible types put 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

} } }

**What will be the output of this loop?**

Output – 0 2 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 } }

**Why does the variable 'x' cause a error? How does scope**

It causes compilation error because ‘x’ is declared inside the for loop. We cannot access ‘x’ outside the for loop.