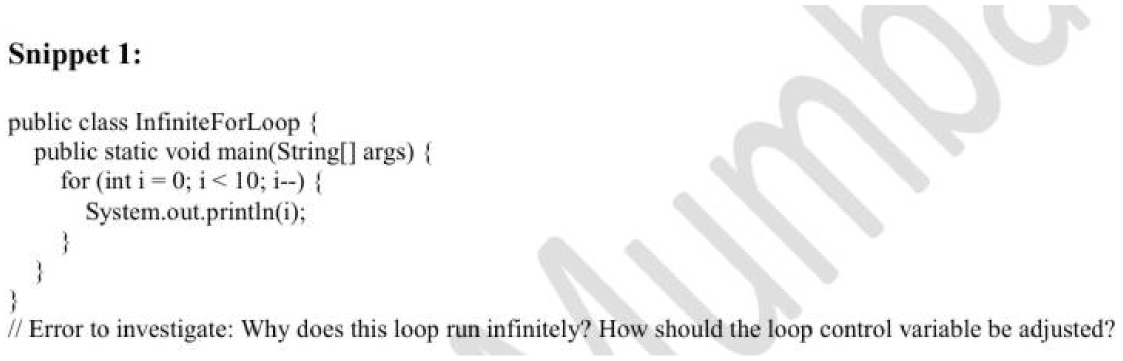
CDAC MUMBAI

Lab Assignment



Snippet 1:

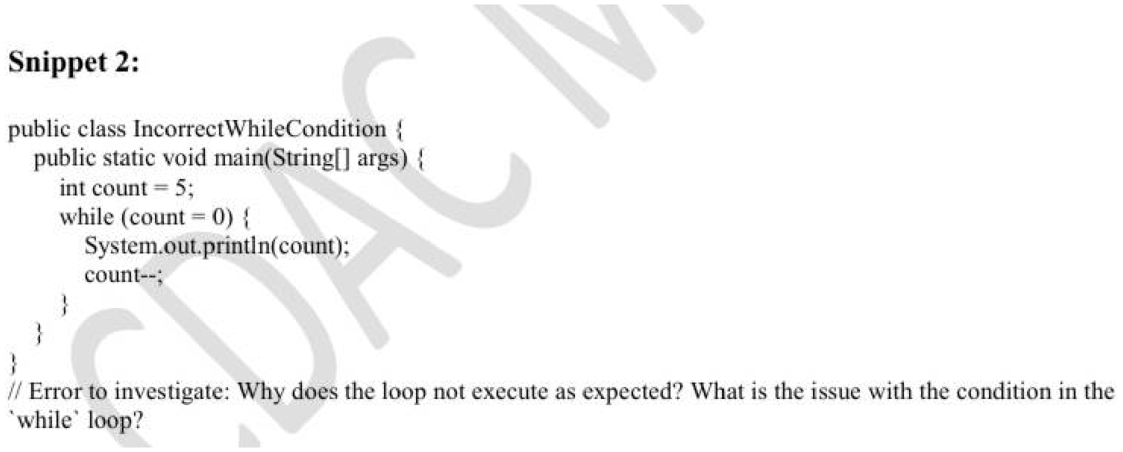
```
public class InfiniteForLoop {
    public static void main(String[] args) {
        for (int i = 0; i < 10; i--) {
            System.out.println(i);
        }
    }
}
// Error to investigate: Why does this loop run infinitely? How should the loop control variable be adjusted?
```

Error:

The loop runs indefinitely because i-- decrements i, making it smaller with each iteration. It will never reach 10, causing an infinite loop.

Correction:

Change i-- to i++:



Snippet 2:

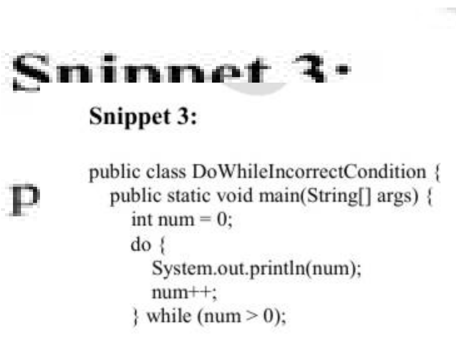
```
public class IncorrectWhileCondition {
    public static void main(String[] args) {
        int count = 5;
        while (count = 0) {
            System.out.println(count);
            count--;
        }
    }
}
// Error to investigate: Why does the loop not execute as expected? What is the issue with the condition in the
`while` loop?
```

Error :

count = 0 is an assignment, not a condition. It always evaluates to 0, which is false, causing the loop to never execute.

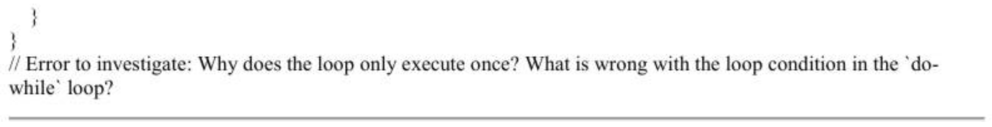
Correction:

Use == instead of =. or change count = 0 to count > 0 .



Snippet 3:

```
public class DoWhileIncorrectCondition {
    public static void main(String[] args) {
        int num = 0;
        do {
            System.out.println(num);
            num++;
        } while (num > 0);
```



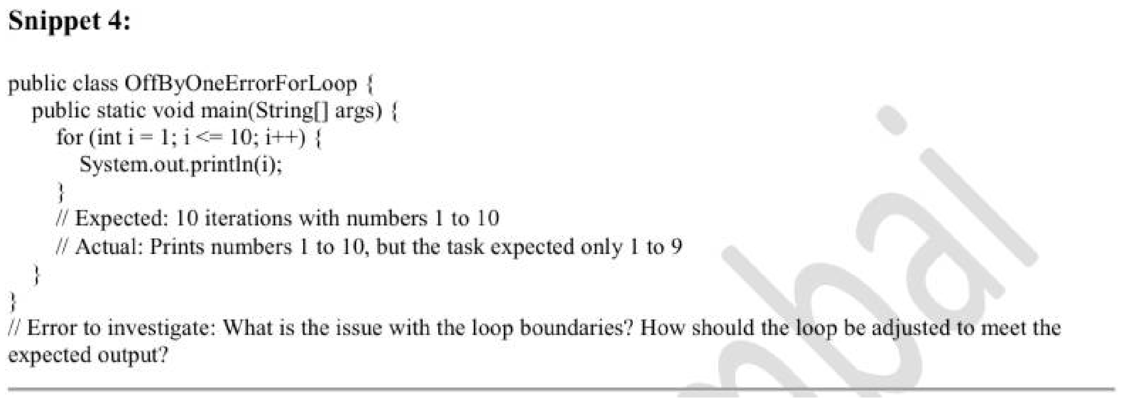
```
    }
}
// Error to investigate: Why does the loop only execute once? What is wrong with the loop condition in the `do-
while` loop?
```

Error:

The loop condition (num > 0) remains true throughout the execution of the code causing it to run infinitely.

Correction:

We have to do multiple fixes the question is wrong. we can change while loop condition to num < 1 rather than num > 0 so it will run once and exit the loop. or make 2 changes to int num = (number greater than 0) and change the do while loop code i.e. num -- instead of num++



Snippet 4:

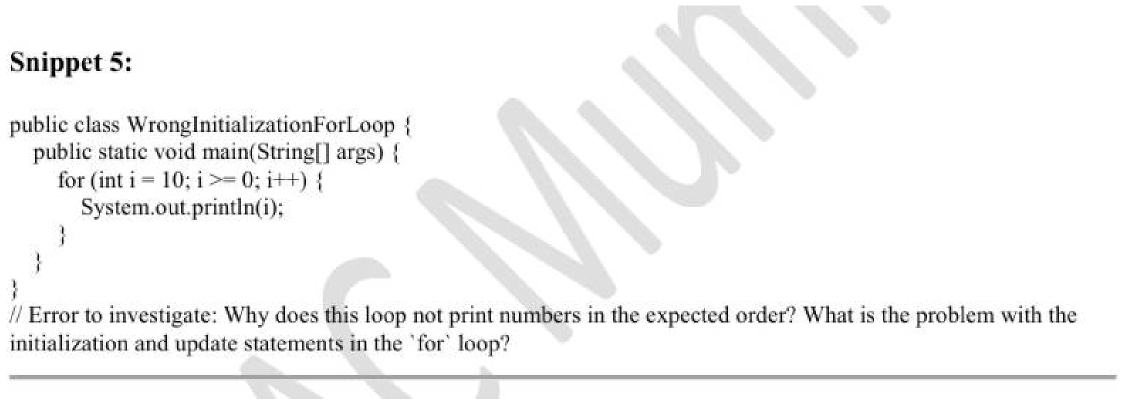
```
public class OffByOneErrorForLoop {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
        // Expected: 10 iterations with numbers 1 to 10
        // Actual: Prints numbers 1 to 10, but the task expected only 1 to 9
    }
}
// Error to investigate: What is the issue with the loop boundaries? How should the loop be adjusted to meet the
expected output?
```

Error:

The loop prints numbers 1 to 10, but the expected output is only 1 to 9.

Correction:

Change loop condition from <= 10 to < 10.


Snippet 5:

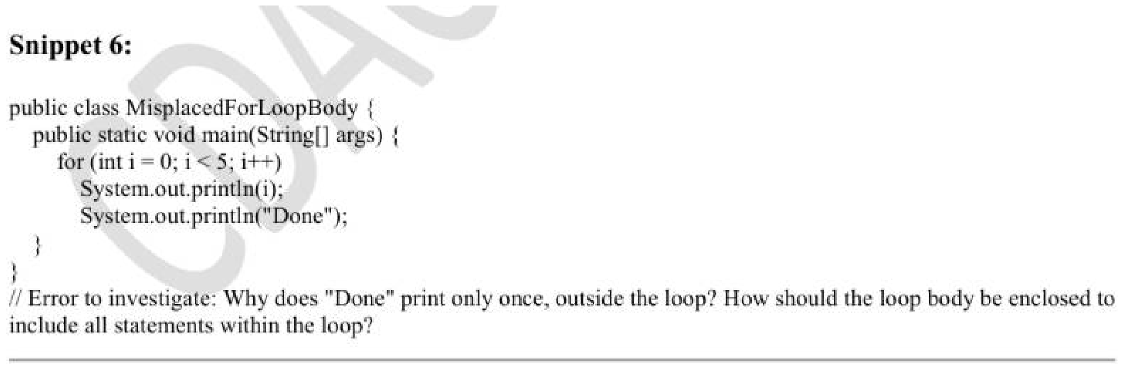
```
public class WrongInitializationForLoop {
    public static void main(String[] args) {
        for (int i = 10; i >= 0; i++) {
            System.out.println(i);
        }
    }
}
// Error to investigate: Why does this loop not print numbers in the expected order? What is the problem with the
initialization and update statements in the `for` loop?
```

Error:

The loop runs infinitely because i++ increases i, making it never >= 0.

Correction:

Use i-- to decrement.


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?
```

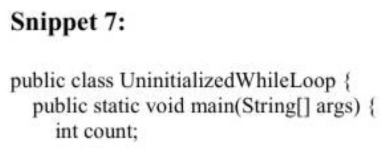
Error:

Done prints only once because it's outside the loop.

Correction:

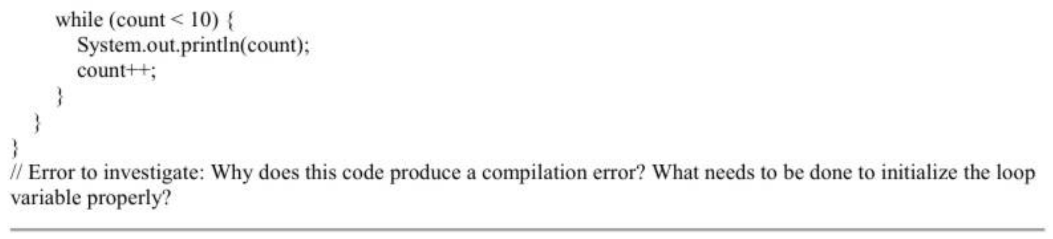
Enclose statements in {}:

```
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?
```

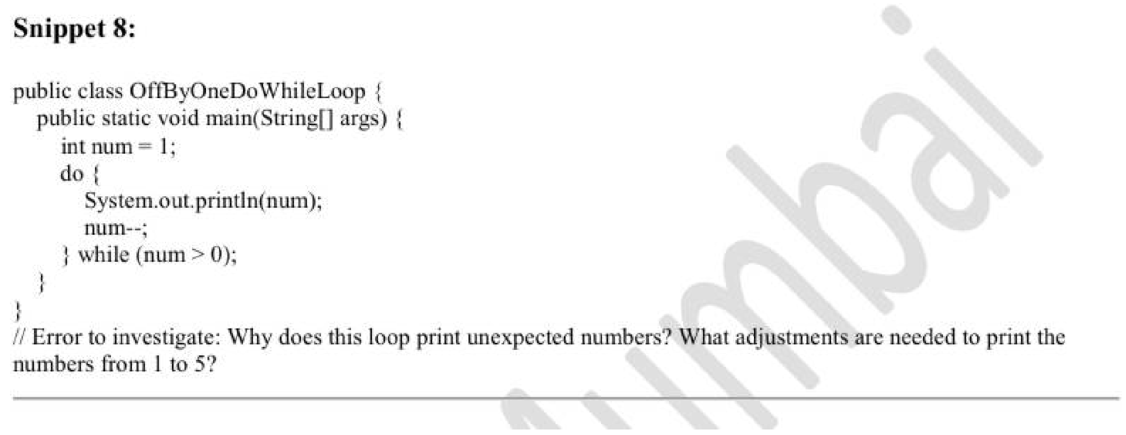
Error:

count is uninitialized, causing a compilation error.

Correction:

Initialize count before using it:

int count = 0;



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?
```

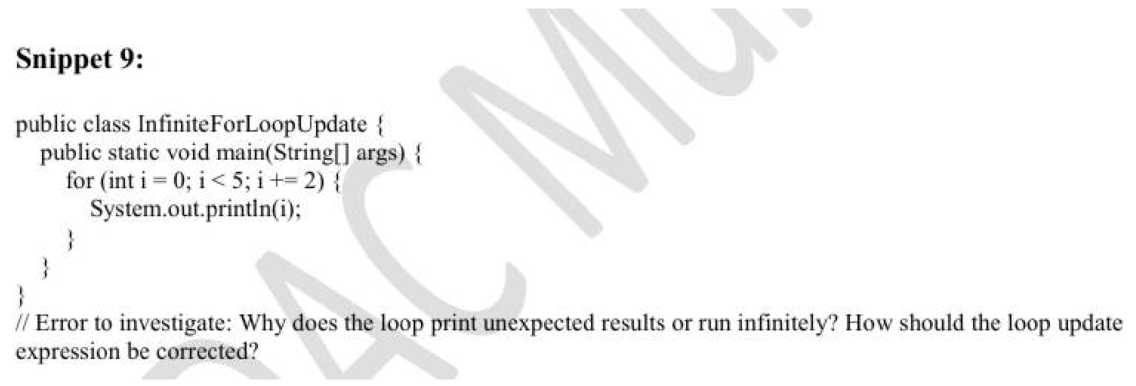
Error:

The loop prints only 1 before exiting.

Correction:

Modify loop to count 1 to 5:

while (num <= 5);


Snippet 9:

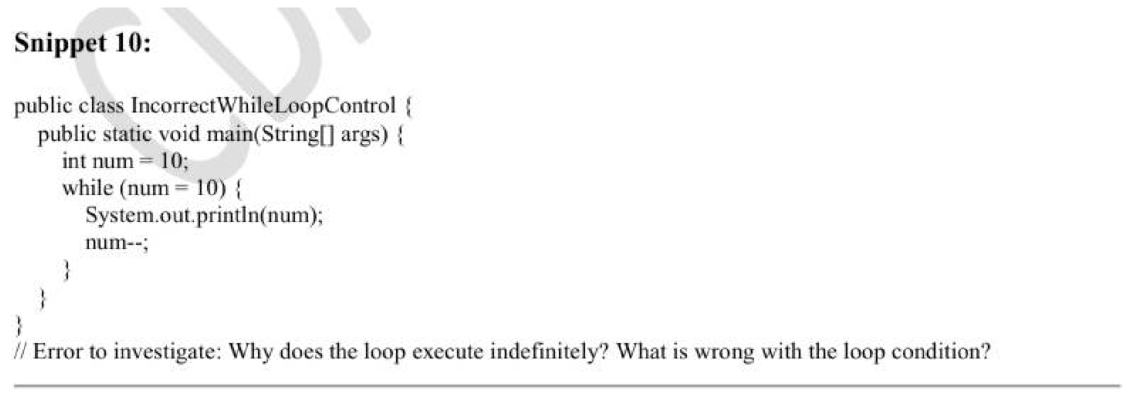
```
public class InfiniteForLoopUpdate {
    public static void main(String[] args) {
        for (int i = 0; i < 5; i += 2) {
            System.out.println(i);
        }
    }
}
// Error to investigate: Why does the loop print unexpected results or run infinitely? How should the loop update
expression be corrected?
```

Error:

No infinite loop, but skips numbers (0, 2, 4 instead of 0, 1, 2, 3, 4).

Correction:

Change increment to i++.


Snippet 10:

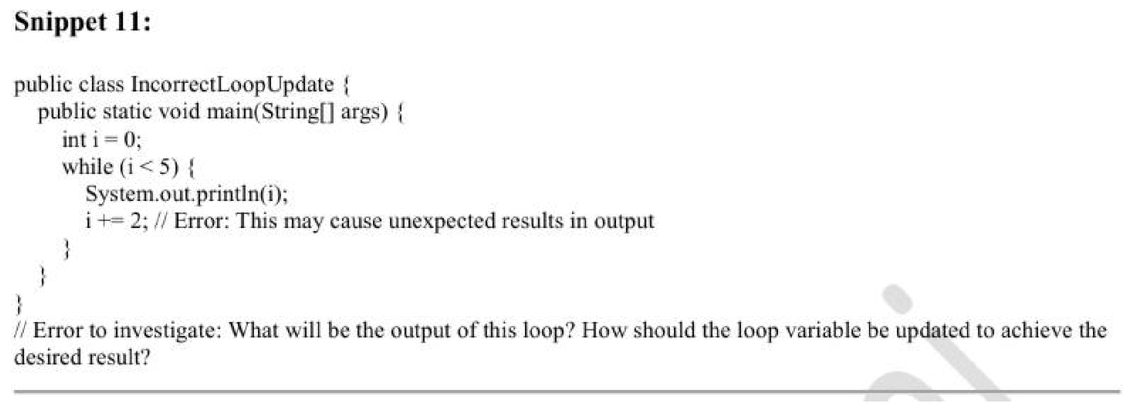
```
public class IncorrectWhileLoopControl {
    public static void main(String[] args) {
        int num = 10;
        while (num = 10) {
            System.out.println(num);
            num--;
        }
    }
}
// Error to investigate: Why does the loop execute indefinitely? What is wrong with the loop condition?
```

Error:

num = 10 is an assignment, not a condition.

Correction:

Use == instead of =:


Snippet 11:

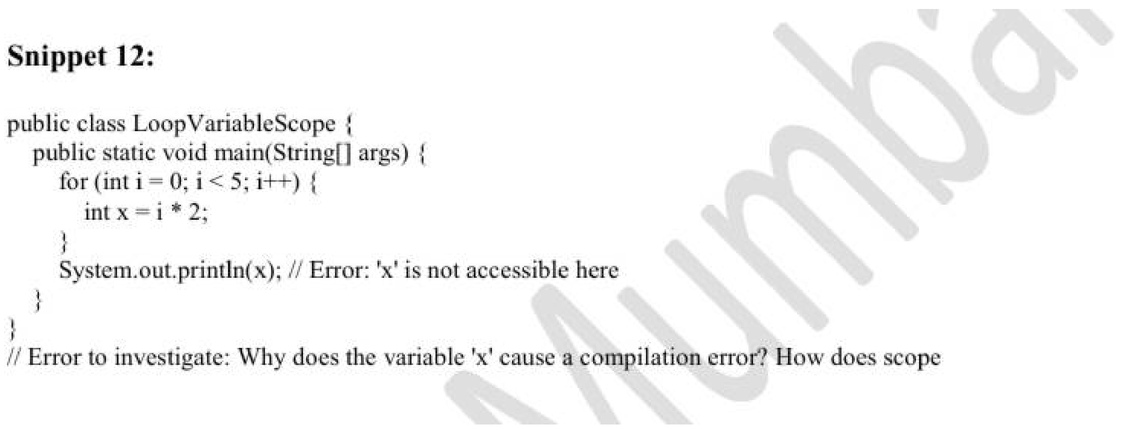
```
public class IncorrectLoopUpdate {
    public static void main(String[] args) {
        int i = 0;
        while (i < 5) {
            System.out.println(i);
            i += 2; // Error: This may cause unexpected results in output
        }
    }
}
// Error to investigate: What will be the output of this loop? How should the loop variable be updated to achieve the
desired result?
```

Error:

Prints 0, 2, 4, missing 1, 3.

Correction:

Change increment to 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? How does scope
```

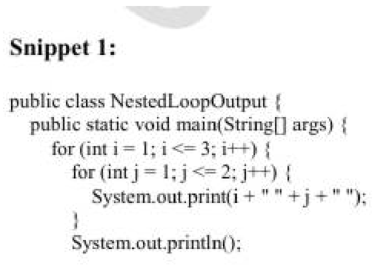
Error:

x is declared inside the loop and goes out of scope after the loop ends.

Correction:

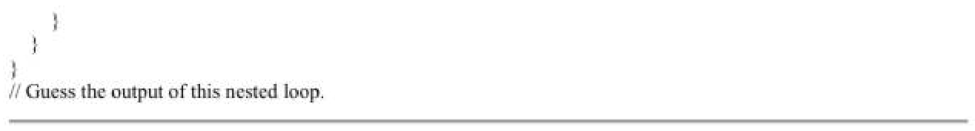
Declare x outside the loop or or print x inside the loop.

SECTION 2: Guess the Output



Snippet 1:

```
public class NestedLoopOutput {
   public static void main(String[] args) {
      for (int i = 1; i <= 3; i++) {
         for (int j = 1; j <= 2; j++) {
            System.out.print(i + " " + j + " ");
         }
         System.out.println();
```



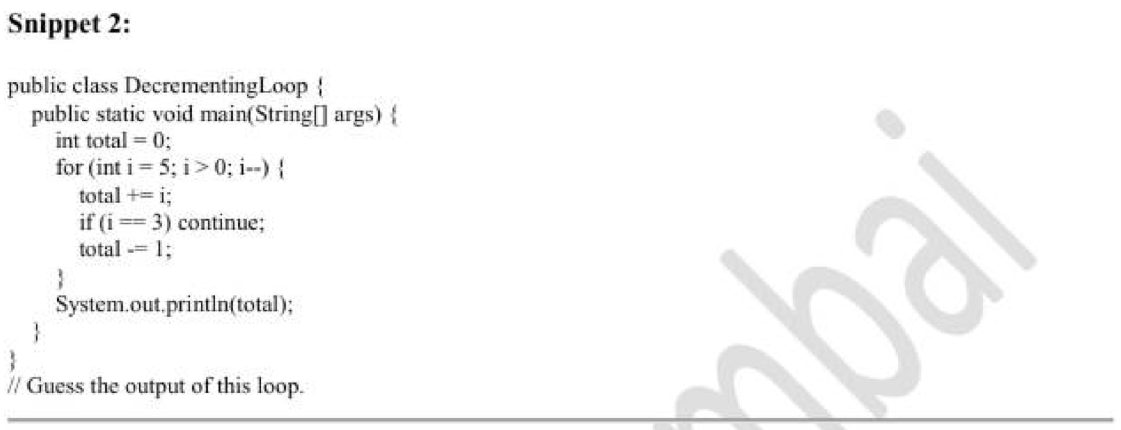
```
      }
   }
}
// Guess the output of this nested loop.
```

Dry Run:

for i = 1, j = 1 → print "1 1 "
for i = 1, j = 2 → print "1 2 " → newline
for i = 2, j = 1 → print "2 1 "
for i = 2, j = 2 → print "2 2 " → newline
for i = 3, j = 1 → print "3 1 "
for i = 3, j = 2 → print "3 2 " → newline

Output :

```
1 1 1 2
2 1 2 2
3 1 3 2
```

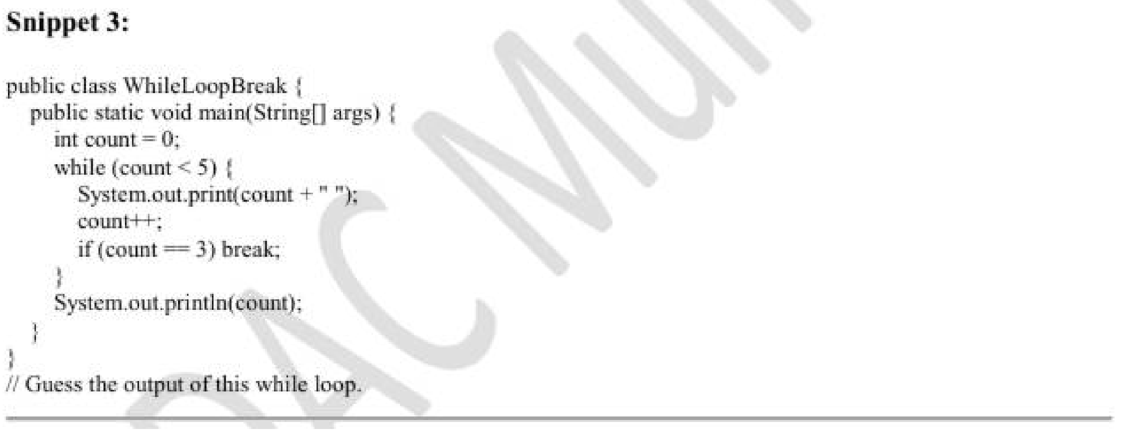


Snippet 2:

```
public class DecrementingLoop {
   public static void main(String[] args) {
      int total = 0;
      for (int i = 5; i > 0; i--) {
         total += i;
         if (i == 3) continue;
         total -= 1;
      }
      System.out.println(total);
   }
}
// Guess the output of this loop.
```

Dry Run:
i = 5: total = 0 + 5 = 5, total -= 1 → 4
i = 4: total = 4 + 4 = 8, total -= 1 → 7
i = 3: total = 7 + 3 = 10 (continue, no total -= 1)
i = 2: total = 10 + 2 = 12, total -= 1 → 11
i = 1: total = 11 + 1 = 12, total -= 1 → 11

Output:
11

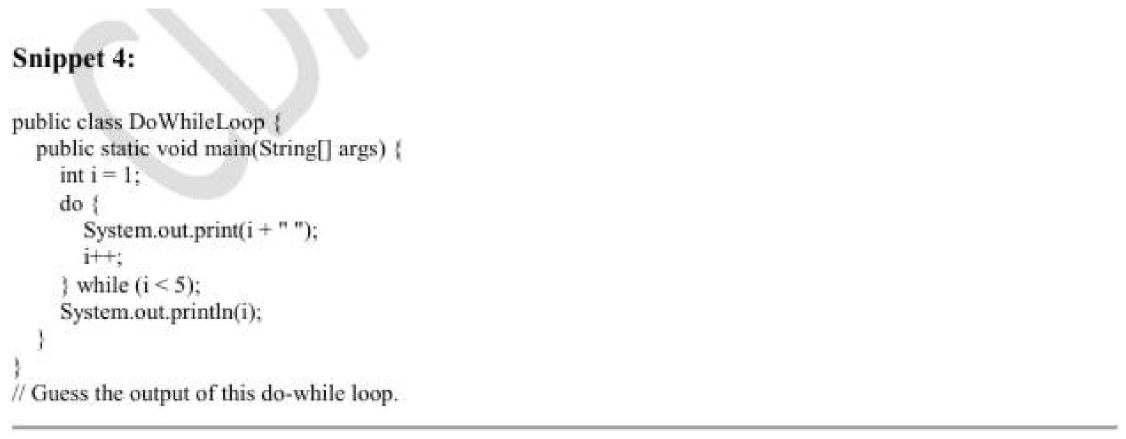


Snippet 3:

```
public class WhileLoopBreak {
   public static void main(String[] args) {
      int count = 0;
      while (count < 5) {
         System.out.print(count + " ");
         count++;
         if (count == 3) break;
      }
      System.out.println(count);
   }
}
// Guess the output of this while loop.
```

Dry Run:
count = 0 → print "0 "
count = 1 → print "1 "
count = 2 → print "2 "
count = 3 → break

Output:
0 1 2 3



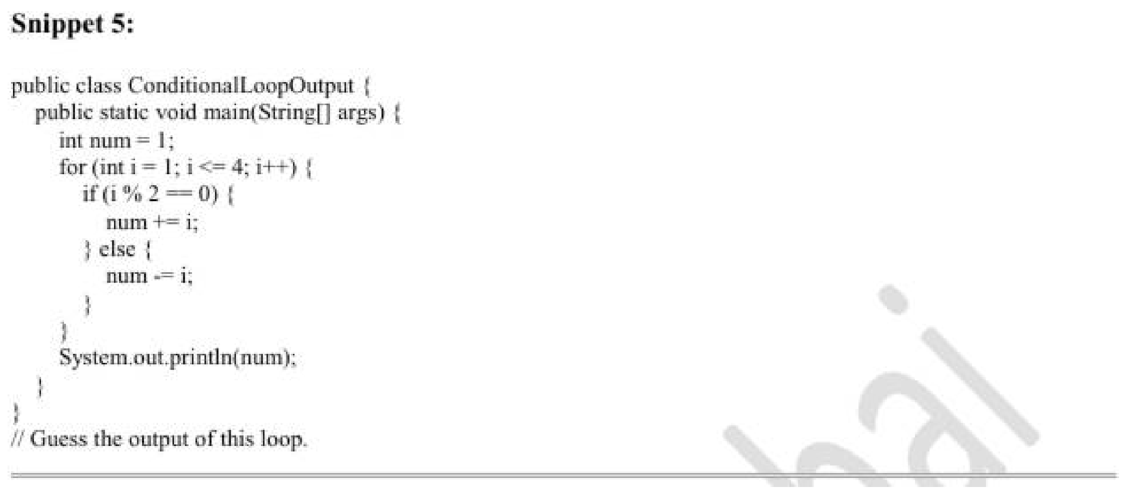
Snippet 4:

```
public class DoWhileLoop {
    public static void main(String[] args) {
        int i = 1;
        do {
            System.out.print(i + " ");
            i++;
        } while (i < 5);
        System.out.println(i);
    }
}
// Guess the output of this do-while loop.
```

Dry Run:
i = 1 → print "1 "
i = 2 → print "2 "
i = 3 → print "3 "
i = 4 → print "4 " (exit loop, i = 5)

Output:

```
1 2 3 4 5
```



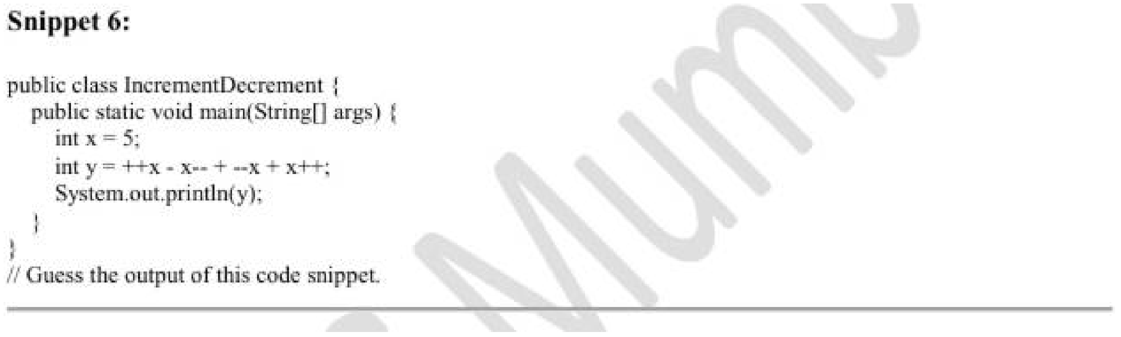
Snippet 5:

```
public class ConditionalLoopOutput {
    public static void main(String[] args) {
        int num = 1;
        for (int i = 1; i <= 4; i++) {
            if (i % 2 == 0) {
                num += i;
            } else {
                num -= i;
            }
        }
        System.out.println(num);
    }
}
// Guess the output of this loop.
```

Dry Run:
i = 1 → num = 1 - 1 = 0
i = 2 → num = 0 + 2 = 2
i = 3 → num = 2 - 3 = -1
i = 4 → num = -1 + 4 = 3

Output:

```
3
```

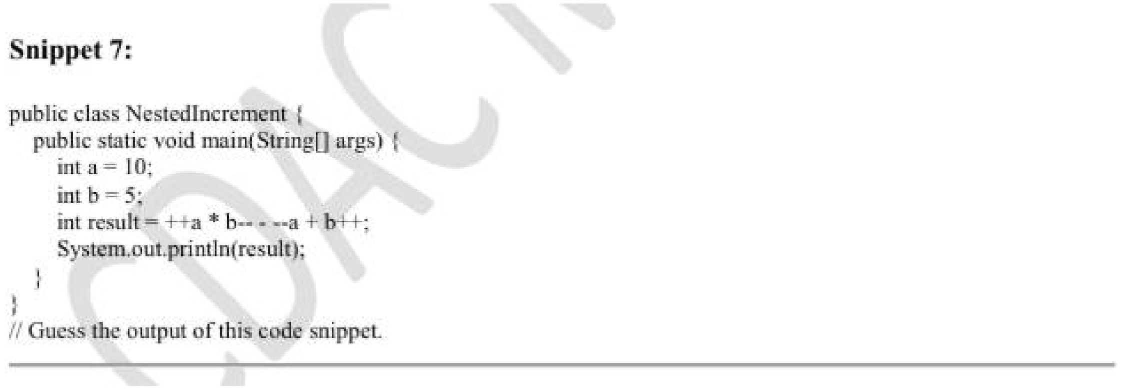


Snippet 6:

```
public class IncrementDecrement {
    public static void main(String[] args) {
        int x = 5;
        int y = ++x - x-- + --x + x++;
        System.out.println(y);
    }
}
// Guess the output of this code snippet.
```

Dry Run:
++x (pre-increment) → x = 6
x-- (post-decrement) → use 6, then x = 5
--x (pre-decrement) → x = 4
x++ (post-increment) → use 4, then x = 5
Calculation: 6 - 6 + 4 + 4 = 8

Output:
8

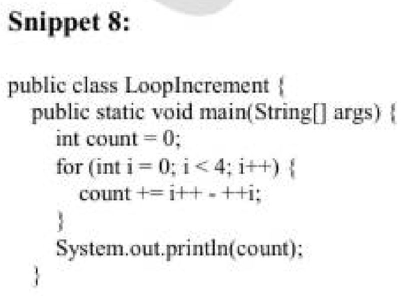


Snippet 7:

```
public class NestedIncrement {
    public static void main(String[] args) {
        int a = 10;
        int b = 5;
        int result = ++a * b-- - --a + b++;
        System.out.println(result);
    }
}
// Guess the output of this code snippet.
```

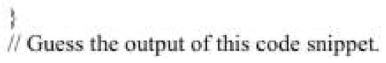
Dry Run:
++a (pre-increment) → a = 11
b-- (post-decrement) → use 5, then b = 4
--a (pre-decrement) → a = 10
b++ (post-increment) → use 4, then b = 5
Calculation: 11 * 5 - 10 + 4 = 55 - 10 + 4 = 49

Output:
49



Snippet 8:

```
public class LoopIncrement {
    public static void main(String[] args) {
        int count = 0;
        for (int i = 0; i < 4; i++) {
            count += i++ - ++i;
        }
        System.out.println(count);
    }
```



```
}
// Guess the output of this code snippet.
```

Dry Run:
i = 0 → count += (0 - 2) = -2 (i becomes 3)
i = 3 → count += (3 - 5) = -2 + (-2) = -4 (i becomes 6, exits loop)

Output :
-4