

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?

Because the condition is getting true for infinite time.

How should the loop control variable be adjusted?

The adjustment of the loop control variable depends on the type of loop and the logic you're implementing.

Correct Code:

```
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? What is the issue with the condition in the `while` loop?

Because in condition count = 0 is given which is consider as integer but it must be Boolean true / false.

Snippet 3:

```
public class DoWhileIncorrectCondition {  
    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 value of num is declared as 0

What is wrong with the loop condition in the `do while` loop?

The do-while loop executes the block of code once before checking the condition. The condition is checked after the loop body has executed. This means the loop body will always run at least once.

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?

The issue with loop boundaries is due to the condition `i <= 10`. This condition allows the loop to continue executing as long as `i` is less than or equal to 10.

How should the loop be adjusted to meet the expected output?

To adjust it the condition should be `i < 10`

Correct Code:

```
public class OffByOneErrorForLoop {  
    public static void main(String[] args) {  
        for (int i = 1; i < 10; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

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?

Because loop is expected to print numbers in descending order starting from 10 down to 0 as `i=10` is declared first.

What is the problem with the initialization and update statements in the `for` loop?

In this code for loop, the initialization statement `int i = 10` sets the loop variable `i` to 10, which is correct. However, the issue is with the update statement `i++`.

Correct Code:

```
public class CorrectInitializationForLoop {  
    public static void main(String[] args) {  
        for (int i = 10; i >= 0; i--) {  
            System.out.println(i);  
        }  
    }  
}
```

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?

Because loop is structured in such a way that Done print only once like until the condition get satisfied it I will get print & then Done will print.

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

The loop body is enclose with curly braces {}

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?

This code produce a compilation error because we have not initialized count before using it in while loop. First the count needs to be initialized and then it should be used in while loop condition.

Correct Code:

```
public class UninitializedWhileLoop {  
    public static void main(String[] args) {  
        int count = 0; // Initialize count while  
        (count < 10) {  
            System.out.println(count);  
            count++;  
        }  
    }  
}
```

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?

The do-while loop in this code prints unexpected numbers because the loop condition is checked after the loop body executes, leading to an off-by-one error.

Correct code:

```
public class OffByOneDoWhileLoop {  
    public static void main(String[] args) {  
        int num = 1; do {  
            System.out.println(num);  
            num--;  
        } while (num >= 0); // Change condition to num >= 0  
    }  
}
```

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?

The loop is not infinite and does not produce unexpected results based on the given code. It correctly prints 0, 2, and 4.

How should the loop update expression be corrected?

The increment expression `i += 2` is appropriate if we want to print numbers starting from 0 and increasing by 2 each time until the condition `i < 5` is false.

Correct code:

```
public class InfiniteForLoopUpdate {  
    public static void main(String[] args) {  
        for (int i = 0; i < 5; i++) {  
            System.out.println(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?

The loop will execute indefinitely due to a mistake in the while loop's condition. The condition used is `num = 10`, which is an assignment operation rather than a comparison operation.

What is wrong with the loop condition?

The thing that is wrong with the loop condition is that assignment operator is used in place of a comparison operator.

Correct Code:

```
public class IncorrectWhileLoopControl {  
    public static void main(String[] args) {  
        int num = 10; 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  
        }  
    }  
}  
  
// Error to investigate: What will be the output of this loop? How should the loop variable be updated to  
// achieve the desired result?
```

The output is:

0

2

4

Correct Code:

```
public class IncorrectLoopUpdate {  
    public static void main(String[] args) {  
        int i = 0; while (i < 5) {  
            System.out.println(i);  
            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

The variable x causes a compilation error because of the scope in which it is declared. x is declared inside the for loop's block this means that x is local to the block of the for loop.

Variables declared inside a block are only accessible within that block. Once the block ends, the variable is no longer in scope and cannot be accessed.

Correct Code:

```
public class LoopVariableScope {
    public static void main(String[] args) {
        int x = 0; // Declare x outside the loop and initialize it
        for (int i = 0; i < 5; i++) {
            x = i * 2;
        }
        System.out.println(x);
    }
}
```

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.
```

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.
```

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.

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.

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.
```

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.
```

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.
```

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.
```

OUTPUT:

-4

SECTION 3: Lamborghini Exercise:

Tasks:

1. Write a program to calculate the sum of the first 50 natural numbers.

CODE:

```
class Natural{
    public static void main(String args[]){
        int sum=0;
        for(int i=0; i<=50; i++){
            sum = sum+i;
        }
        System.out.println("Sum of the first 50 natural numbers is " + sum);
    }
}
```

OUTPUT:

Sum of the first 50 natural numbers is 1275

2. Write a program to compute the factorial of the number 10.

CODE:

```
public class Factorial{
    public static void main(String args[]){

        int i, fact=1;
        int num=10;
        for(i=1;i<=num;i++){
            fact=fact*i;
        }
        System.out.println("Factorial of "+num+" is: "+fact);
    }
}
```

OUTPUT:

Factorial of 10 is: 3628800

3. Write a program to print all multiples of 7 between 1 and 100.

CODE:

```
class Multiples{
    public static void main(String args[]){
```

```

        System.out.print("All multiples of 7 between 1 and 100 are ");
        for(int i=1; i<=100; i++){
            if(i%7==0){
                System.out.print(i+" ");
            }
        }
    }
}

```

OUTPUT:

All multiples of 7 between 1 and 100 are 7 14 21 28 35 42 49 56 63 70 77 84 91 98

4. Write a program to reverse the digits of the number 1234. The output should be 4321.

CODE:

```

public class Reverse{
    public static void main(String args[]){
        int num = 1234;
        int reversedNum = 0;

        while (num != 0) {
            int rem = num % 10;
            reversedNum = reversedNum * 10 + rem;
            num = num / 10;
        }

        System.out.println("Reversed Number is: " + reversedNum);
    }
}

```

OUTPUT:

Reversed Number is: 4321

5. Write a program to print the Fibonacci sequence up to the number 21.

CODE:

```

public class FibonacciSequence {
    public static void main(String[] args) {
        int num1 = 0, num2 = 1;

        System.out.print("Fibonacci Sequence up to 21: " + num1 + ", " + num2);
    }
}

```

```

int nextNum = num1 + num2;

while (nextNum <= 21) {
    System.out.print(", " + nextNum);

    num1 = num2;
    num2 = nextNum;
    nextNum = num1 + num2;
}
}
}

```

OUTPUT:

Fibonacci Sequence up to 21: 0, 1, 1, 2, 3, 5, 8, 13, 21

6. Write a program to find and print the first 5 prime numbers.

CODE:

```

public class PrimeNumber {
    public static void main(String[] args) {
        int num = 5;
        int count;
        System.out.print("The first 5 prime numbers are: ");

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

            for (int j = 2; j <= i / 2; j++) {
                if (i % j == 0) {
                    count++;
                    break;
                }
            }

            if (count == 0) {
                System.out.print(i + " ");
            }
        }
    }
}

```


OUTPUT:

The first 5 prime numbers are: 1 2 3 5

7. Write a program to calculate the sum of the digits of the number 9876. The output should be 30 (9 + 8 + 7 + 6).

CODE:

```
public class Sum {  
    public static void main(String[] args) {  
        int num = 9876;  
        int sum = 0;  
  
        while (num > 0) {  
            int digit = num % 10;  
            sum += digit;  
            num = num / 10;  
        }  
        System.out.println(sum + "(9 + 8 + 7 + 6)");  
    }  
}
```

OUTPUT:

30(9 + 8 + 7 + 6)

8. Write a program to count down from 10 to 0, printing each number.

CODE:

```
public class Countdown {  
    public static void main(String[] args) {  
        int count = 10;  
  
        while (count >= 0) {
```

```
        System.out.println(count);

        count--;
    }
}
}
```

OUTPUT:

```
10
9
8
7
6
5
4
3
2
1
0
```

9. Write a program to find and print the largest digit in the number 4825.

CODE:

```
public class LargestDigit {
    public static void main(String[] args) {
        int num = 4825;
        int larDigit = 0;

        while (num > 0) {
            int digit = num % 10;

            if (digit > larDigit) {
                larDigit = digit;
            }
            num = num / 10;
        }
        System.out.println("The largest digit is: " + larDigit);
    }
}
```

OUTPUT:

The largest digit is: 8

10. Write a program to print all even numbers between 1 and 50.

CODE:

```
public class EvenNumber{
    public static void main(String args[]){
        System.out.print("All even numbers between 1 and 50 are ");
        for(int i=1; i<=50; i++){
            if(i%2==0){
                System.out.print(i+" ");
            }
        }
    }
}
```

OUTPUT:

All even numbers between 1 and 50 are 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34
36 38 40 42 44 46 48 50

11. Write a Java program to demonstrate the use of both pre-increment and post-decrement operators in a single expression

CODE:

```
import java.util.*;

public class IncDec{

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a");

        int a = sc.nextInt();

        System.out.println("Enter b");

        int b = sc.nextInt();

        int result;
```

```

        result = ++a + b--;

        System.out.println("The result is "+result);
    }
}

```

OUTPUT:

Enter a

11

Enter b

15

The result is 27

12. Write a program to draw the following pattern:

```

*****
*****
*****
*****
*****

```

CODE:

```

public class Pattern1{

    public static void main(String args[]){

        int row = 5;

        int col = 5;

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

            for(int j=0; j<col; j++){

```

```

        System.out.print("*");
    }
    System.out.println(" ");
}
}
}

```

13. Write a program to print the following pattern:

```

1
2*2
3*3*3
4*4*4*4
5*5*5*5*5
5*5*5*5*5
4*4*4*4
3*3*3
2*2 1

```

CODE:

```

public class Pattern2{
    public static void main(String args[]){
        int row=5;

        for(int i=1; i<=row; i++){
            for(int j=1; j<=i; j++){
                System.out.print(i);
                if(j<i){
                    System.out.print("*");

```

```

        }
    }
    System.out.println();
}

for(int i=row; i>=2; i--){
    for(int j=1; j<i; j++){
        System.out.print(i-1);
        if(j<i-1){
            System.out.print("*");
        }
    }
    System.out.println();
}
}
}

```

14. Write a program to print the following pattern:

```

*

**

***

****

*****

*****

*****

```

CODE:

```

public class Pattern3{
    public static void main(String[] args){

```

```

        for (int i = 1; i <= 9; i += 2){
            for (int j = 1; j <= i; j++){
                System.out.print("*");
            }
            System.out.println();
        }
    }
}

```

15. Write a program to print the following pattern:

```

*
**
***
****
*****

```

CODE:

```

public class Pattern4{
    public static void main(String args[]){

        for(int i=1; i<=5; i++){
            for(int j=0; j<i; j++){
                System.out.print("*");
            }
            System.out.println(" ");
        }
    }
}

```

16. Write a program to print the following pattern:

```
*  
  
***  
  
*****  
  
*****  
  
*****
```

CODE:

```
public class Pattern5{  
    public static void main(String args[]) {  
  
        for (int i = 1; i <= 9; i+=2) {  
  
            for (int j = 1; j <=i; j++) {  
                System.out.print("*");  
            }  
  
            System.out.println();  
        }  
    }  
}
```

17. Write a program to print the following pattern:

```
*****  
  
****  
  
***
```

CODE:

```
public class Pattern6{  
    public static void main(String args[]) {  
  
        for (int i = 5; i>=1; i--) {  
  
            for (int j = 1; j <= i; j++) {  
                System.out.print("* ");  
            }  
  
            System.out.println();  
        }  
    }  
}
```

18. Write a program to print the following pattern:

CODE:

```
public class Pattern7{
```

```

    public static void main(String args[]) {

        for (int i = 1; i <=7; i+=2) {

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

                System.out.print("*");

            }

            System.out.println();

        }

        for (int i = 5; i >= 1; i-=2) {

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

                System.out.print("*");

            }

            System.out.println();

        }

    }
}

```

19. Write a program to print the following pattern:

```

1
1*2
1*2*3
1*2*3*4
1*2*3*4*5

```

CODE:

```
public class Pattern8{  
    public static void main(String args[]){  
        for(int i=1; i<=5; i++){  
            for(int j=1; j<=i; j++){  
                System.out.print(j);  
                if(j<i){  
                    System.out.print("*");  
                }  
            }  
            System.out.println("");  
        }  
    }  
}
```

20. Write a program to print the following pattern:

```
5  
5*4  
5*4*3  
5*4*3*2  
5*4*3*2*1
```

CODE:

```
public class Pattern9{  
    public static void main(String args[]){  
        for(int i=5; i>=1; i--){  
            for(int j=5; j>=i; j--){  
                System.out.print(j);
```

```

        if(j>i){
            System.out.print("*");
        }
    }
    System.out.println();
}
}
}

```

21. Write a program to print the following pattern:

```

1
1*3
1*3*5
1*3*5*7
1*3*5*7*9

```

CODE:

```

public class Pattern10{
    public static void main(String args[]){
        for(int i=1; i<=5; i++){
            int num=1;
            for(int j=1; j<=i; j++){
                System.out.print(num);
                num+=2;
                if(j<i){
                    System.out.print("*");
                }
            }
        }
    }
}

```

```

        System.out.println();
    }
}

```

22. Write a program to print the following pattern:

```

*****
*****
*****
***
*
***
*****
*****
*****

```

CODE:

```

public class Pattern11{
    public static void main(String args[]){
        int row= 5;
        for(int i=row; i>=1; i--){
            for(int j=0; j<row-1; j++){
                System.out.print("");
            }
            for(int j=0; j<(2*i-1); j++){
                System.out.print("*");
            }
            System.out.println();
        }
    }
}

```

```

    }

    for(int i=2; i<=row; i++){
        for(int j=0; j<row-1; j++){
            System.out.print("");
        }
        for(int j=0; j<(2*i-1); j++){
            System.out.print("*");
        }
        System.out.println();
    }
}

```

23. Write a program to print the following pattern:

11111

22222

33333

44444

55555

CODE:

```

public class Pattern12{
    public static void main(String args[]){
        for(int i=1; i<=5; i++){
            for(int j=1; j<=5; j++){
                System.out.print(i);
            }

```

```

        System.out.println();
    }
}

```

24. Write a program to print the following pattern:

```

1
22
333
4444
55555

```

CODE:

```

public class Pattern13{
    public static void main(String args[]){
        for(int i=0; i<=5; i++){
            for(int j=1; j<=i; j++){
                System.out.print(i);
            }
            System.out.println();
        }
    }
}

```

25. Write a program to print the following pattern:

```

1
12
123

```

1234

12345

CODE:

```
public class Pattern14{  
    public static void main(String args[]){  
        for(int i=0; i<=5; i++){  
            for(int j=1; j<=i; j++){  
                System.out.print(j);  
            }  
            System.out.println();  
        }  
    }  
}
```

26. Write a program to print the following pattern: 1

2 3

4 5 6

7 8 9 10

CODE:

```
public class Pattern15{  
    public static void main(String args[]){  
        int row= 1;  
        for(int i=0; i<=5; i++){  
            for(int j=1; j<=i; j++){  
                System.out.print(row + " ");  
                row++;  
            }  
        }  
    }  
}
```



```
System.out.println();
```

```
}
```

```
}
```

```
}
```