Snippet 1

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

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

                        System.out.println(i);

} }

}
```

Ans : - Here the error is in updation block , because if (i- -) then it should start with 10 and it should go in loop until (i >= 0);

OR we can simply change the updation block to (i + +)

Snippet 2

```
public class IncorrectWhileCondition {

        public static void main(String[] args) {

                int count = 5;

                while (count = 0) {

                        System.out.println(count);

                        count--; }

} }
```

Ans : - Here the error is in condition block where the (count = 0) means its a assignment operator. Instead we should use (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);

        }

}
```

Ans :- In this code error is in conditional statement and this will run to infinite loop because of num > 0 , which will be true because in do loop it will get increase to 1 which is greater than zero hence it will go to infinity.

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

} }
```

Ans :- To print the number from 1 to 9 we have to change the conditional block to I < 10 because if I <= 10 then it will print till the 10 because of "<=" sign.

Snippet 5

```
public class WrongInitializationForLoop {

        public static void main(String[] args) {

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

                        System.out.println(i);

        } }

}
```

Ans :- Here the problem is in updation block because if we want to print number from 10 to 0 then we should use (i - -) instead of (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");
} }
```

Ans :- Here the immediate statement after the for loop will be considered in the loop . If we want to make Done statement in the loop then we have to enclose both the statement with“{}“.

Snippet 7

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

Ans :- The problem in the code is count is not initialise with any value. In java initialisation statement must be declared with any value.

Snippet 8

```
public class OffByOneDoWhileLoop {
         public static void main(String[] args) {
                int num = 1;
                do {
                        System.out.println(num);
```

```
            num--;
        } while (num > 0);

} }
```

Ans :- loop will run only once because num- - make num value from 1 to 0 and make conditional statement false so it will only execute one time . So if we want to print number from 1 to 5 then we should change conditional statement to (num >= 5). Which will run loop till num value become 5 .And we have to make num - - to num + +.

Snippet 9

```
public class InfiniteForLoopUpdate {

        public static void main(String[] args) {

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

                System.out.println(i);

            } }

}
```

Ans :- The code will give the output as 0 , 2 ,4 because in every iteration I value increased by 2 and till I < 5 there are only 3 numbers to be printed. To Avoid infinite loop we should use updation statement.

Snippet 10

```
public class IncorrectWhileLoopControl {

    public static void main(String[] args) {

        int num = 10;

        while (num = 10) {

            System.out.println(num);

            num--; }

} }
```

Ans :- Here there is error in conditional statement because num = 10 is assigning the value of 10 , but if we use == operator then it will campare num value with 10.

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 }
} }
```

Ans :- The code will give the output as 0 , 2 ,4 because in every iteration I value increased by 2 and till I < 5 there are only 3 numbers to be printed. To Avoid infinite loop we should use updation statement.

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 }
}
```

Ans :- Here X is not accessible because SOP statement of x is outside the scope of variable x.