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?

ans=in that incremenation is not given properly thats why loop goes in infinite time.

How should the loop control variable be adjusted?

ans=in the for loop in code i-- given instead of that write i++.code will sucessfully.

correct code:

```
public class InfiniteForLoop {
 public static void main(String[] args) {
 for (int i = 0; i < 10; i++) {
 System.out.println(i);
 }
 }
} //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)//error
 {
 System.out.println(count);
 count--;
```

```
 }
 }
}
```

Error:

incompatible types: int cannot be converted to boolean

```
 while (count = 0) {
```

Que:Why does the loop not execute as expected?

ans=because in the while loop condition is not give properly in while count=0 int cannot be converted to boolean invalid because 0 is not a boolean in Java.

Que:What is the issue with the condition in the while loop?

ans=in the while loop int is not converted into boolean so in while give the condition while(count>=0) now code will run sucessfully.

Correct code

```
public class IncorrectWhileCondition {
 public static void main(String[] args) {
 int count = 5;
 while (count >= 0)
 {
 System.out.println(count);
 count--;
 }
 }
}//5 4 3 2 1 0
```

//snnipt3

```
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? What is wrong with the loop condition in the `dowhile` loop?

Snnipt 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?

ans=Their is no issue in the program run sucessfully but we want 1 to 9 output.so we need change the condtion (i<10) then program give 1 to 9 output.

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

ans=we need change the condtion of for loop (int i=1;i<10;i++)(remove the <=) we write this code execute 1 to 9.

Correct code

```
public class OffByOneErrorForLoop {
 public static void main(String[] args) {
 for (int i = 1; i < 10; i++)//remove <= from this
 {
 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
 }
}
```

//Snippet 5:

```
public class WrongInitializationForLoop {
 public static void main(String[] args) {
 for (int i = 10; i >= 0; i++)//error
 {
 System.out.println(i);
 }
 }
}//goes in infinte loop
```

Why does this loop not print numbers in the expected order?

ans= beacuse incremention is give proper that's why error occur give i-- instaed of i++.

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

ans=no problem with initialization but we change the update statement write i-- instead of i++.

Correct Code:

```
public class WrongInitializationForLoop {
```

```
 public static void main(String[] args) {
 for (int i = 10; i >= 0; i--)
 {
 System.out.println(i);
 }
 }
}//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");
 }
} //0 1 2 3 4 Done
```

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

ans=In Java, if a for loop does not use {} braces, only the first statement after it is considered inside the loop.suppose after for loop i give {

brace then second sop statement inside the loop that's why System.out.println("Done"); print only ones

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

ans=use the {} brace after for loop

Correct Code:

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

```
 System.out.println("Done");
}
 }
} //0 Done 1 Done 2 Done 3 Done 4 Done
```

Snippet 7:

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

Que

Why does this code produce a compilation error?

ans=variable count might not have been initialized that's why compilation error.we must initialized value.

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

ans=give the proper initializtion int count=10; like that.

Correct code:

```
public class UninitializedWhileLoop {
 public static void main(String[] args) {
 int count=1;
 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);
 }
}
```

Why does this loop print unexpected numbers?

ans= do-while loop prints only 1 because:

The loop starts with num = 1.

It prints 1.Then, num-- decrements num to 0.The condition while (num > 0) fails, so the loop terminates.

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

ans=we need give :

1.give incremention i++ instead of i-- correctly.

2.in while give proper condtion (num<=5)

we want print 1 to 5.

correct code:

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

```
int num = 1;
do {
System.out.println(num);
num++;
} while (num <= 5);
}
}
```

//Snnpit 9

```
public class InfiniteForLoopUpdate {
 public static void main(String[] args) {
 for (int i = 0; i < 5; i += 2) {
 System.out.println(i);
 }
 }
}
```

Why does the loop print unexpected results or run infinitely?

ans=it will print 0 2 4 because:

1.first i=0 then 0<5 is correct i+=2=i+2=0+2=2 value of i become 2

2.again i=2 then 2<5 is correct i+2=2+2=4 value of i become 4

3.now value of i=4 then 4<5 is correct but i+2=4+2 now value is 6 is not correct loop get terminated.

How should the loop update expression be corrected?

ans=yes is corrected it print 0 2 4

Snippet 10:

```
public class IncorrectWhileLoopControl {
 public static void main(String[] args) {
 int num = 10;
 while (num = 10) {
```

```
System.out.println(num);
num--;
}
}
}
```

Why does the loop execute indefinitely?

ans=: incompatible types: int cannot be converted to boolean

while (num = 10) in that condition given incorrectly their is an incompatiable type error.

What is wrong with the loop condition?

ans=i that code

1.initialization is not given properly we need give i=1

2.in while condtion while(num<=10)

3.give num++ instead of i--

correct code:

```
public class IncorrectWhileLoopControl {
public static void main(String[] args) {
int num = 1;
while (num <= 10) {
System.out.println(num);
num++;
}
}
} //1 2 3 4  5 6 7 8 9 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
}
}
}
```

What will be the output of this loop?

ans=There is no syntax error in this code, but the issue is with the loop update (i += 2;), which skips numbers and does not print sequential values from 0 to 4.

output of this is 0 2 4 it skip the 1 3 beacuse of i+=2 =i+2

How should the loop variable be updated to achieve the

desired result?

ans=so need to change the incremention i++ instead of i+=2

Corrct code:

```
public class IncorrectLoopUpdate {
 public static void main(String[] args) {
 int i = 0;
 while (i < 5) {
 System.out.println(i);
 i++;
 }
} //0 1 2 3 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 compilation error?

ans=cannot find symbol System.out.println(x);in the for loop we give i and we assigning x in that so it will not corrrct thats why showing symbol not found.

How does scope?

ans=we need to give x outside the loop.

Correct code:

```
public class LoopVariableScopeFixed {

public static void main(String[] args) {

int x = 0; // Declare x outside the loop

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

x = i * 2;

}

System.out.println(x); //

}
```

^