1] /\*public class D3S1 {

public static void main(String[] args) {

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

System.out.println(i);

}

} }

infinite loop due to i is not incrementing instead it get decremented and the output is in negative integers\*/

**correct program:**

public class D3S1 {

public static void main(String[] args) {

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

System.out.println(i);

}

} }

2] public class D3S2 {

public static void main(String[] args) {

int count = 5;

while (count = 0) {

System.out.println(count);

count--;

}

}

}

/\* ERROR: incompatible types: int cannot be converted to boolean

this is because the condition in while loop must be in the form of Boolean expression that is the condition evaluation will be either true or false \*/

**correct program**

public class D3S2 {

public static void main(String[] args) {

int count = 5;

while (count>0) {

System.out.println(count);

count--;

}

}

}

3] public class DoWhileIncorrectCondition {

public static void main(String[] args) {

int num = 0;

do {

System.out.println(num);

num++;

} while (num > 0);

}

}

Infinite loop because in do’s body num is incrementing by 1 and in the while condition ,the condition is getting true that is num>0. Hence we get infinite loop. Slight change in code can eliminate the infinite loop

public class DoWhileIncorrectCondition {

public static void main(String[] args) {

int num = 0;

do {

System.out.println(num);

num++;

} while (num < 10); }

}

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?

By changing the condition part of for loop to for(int i=1; i<10;i++) we can achieve the desired output

public class OffByOneErrorForLoop {

public static void main(String[] args) {

for (int i = 1; i < 10; i++) {

System.out.println(i);

}

}

}

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?

The output of the above code is infinite loop because the value of i is incrementing in every iteration and satisfying the condition . It will never dissatisfy and come out off the loop. To get the output from 10….0 we change the post inc by post dec

public class WrongInitializationForLoop {

public static void main(String[] args) {

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

System.out.println(i);

}

}

}

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?

The "Done" print only once, because the scope of the for loop is upto the first “;”. Therefore, the sop statement is out of the scope of the for loop. Now the below code include all statements in loop

public class MisplacedForLoopBody {

public static void main(String[] args) {

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

System.out.println(i);

System.out.println("Done");

}

} }

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: variable count might not have been initialized.

We need to initialized the count variable to get desired output . The below code will give the output as 5 6 7 8 9

public class D3S2 {

public static void main(String[] args) {

int count=5;

while (count < 10) {

System.out.println(count);

count++;

}

}

}

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 output of above program will be 1 because we are doing num --so 1 is decreased by 1 that is it will become 0 and then checking it inside in the while loop , so here the condition gets false the control is transfer outside the loop. To get the ouput like 1…5 we need to do like

public class OffByOneDoWhileLoop {

public static void main(String[] args) {

int num = 1;

do {

System.out.println(num);

num++;

} while (num <= 5);

}

}

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?

The code is working correctly and giving output like 0 2 4 because the values of I are incremented by i=i+2.

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: incompatible types: int cannot be converted to Boolean

while (num = 10) here we are assigning 10 to num variable. While loops condition must to be evaluated to true or false. To print 10……1 we are going to doing some changes like

public class IncorrectWhileLoopControl {

public static void main(String[] args) {

int num = 10;

while (num >0) {

System.out.println(num);

num--;

}

} }

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

}

}

}

The code works correctly, and gives the output like 0 2 4.

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: The error occurs because the variable x is declared inside the for loop, making it local to the loop's block. A variable's scope is limited to the block in which it is declared. The correct program is

public class LoopVariableScope {

public static void main(String[] args) {

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

int x = i \* 2;

System.out.print(x);

}

}

}

The output is 0 2 4 6 8

**Guess the Output**

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();

}

}

}

Output:

1 1 1 2

2 1 2 2

3 1 3 2

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);

}

}

//I= 5-4-3-2-1

//T=0/5/4-8/7-10-12/11-12/11

Output:

11

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);

}

}

//count=0/1/2/3

//sop=0/1/2/3—count ==3 break control is out of the loop

Output is 0 1 2 3

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);

}

}

//i=1/2/3/4/5 🡪5<5 -F out of loop -> sop displays the current value of I ->5

//sop=1-2-3-4-5

Output:

1 2 3 4 5

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);

}

}

//num=1/0/2/-1/3

//i=1/2/3/4

1%2=1 | 2%2=0 | 3%2=1 |4%2=0

Output : 3

6] public class IncrementDecrement {

public static void main(String[] args) {

int x = 5;

int y = ++x - x-- + --x + x++;

System.out.println(y);

}

}

//x=5/6/5/4/5

//y=6-6+4+4=8

Output:

8

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);

}

}

//a=10/11/10

//b=5/4/5

//result=11\*5-10+4=49

Output:

49

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);

}

}

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);

}

}

//i=0/1/2/3/4/5/6🡪6<4 🡪F

//count=0/-2/

Count + i++ - ++i

0+0-2 = -2

-2+3-5=-4

Output :

-4