Assignment 3 SECTION 1 : Loop Errors

Snippet 1:

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

Output of the code runs infinitely because instead of increasing i towards 10, the loop decreases.

Corrected 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--;
              }
       }
}
```

Error:

```
IncorrectWhileCondition.java:4: error: incompatible types: int cannot be con
verted to boolean
while (count = 0) {
             ^
1 error
```

while loop has checked that count should be 0 but here count is initialized to 5, so it is not going inside while loop.

Corrected code

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

Snippet 3:

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

The given code runs infinitely because num is initialized to 0, and the do-while loop condition is num > 0. Since num is incremented in each iteration and never decreases, the condition always remains true, resulting in an infinite loop.

Corrected code:

```
public class DoWhileIncorrectCondition {
        public static void main(String[] args) {
        int num = 0;
        do {
                System.out.println(num);
                num++;
        }while (num < 5);
    }
}
```

Snippet 5:

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

The given code runs infinitely because i is initialized to 10, and the for-loop condition is num >= 0. Since i is incremented in each iteration and never decreases, resulting in an infinite loop.

Corrected code:

```
public class WrongInitializationForLoop {
        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");
    }
}
```

For loop does not have opening and closing bracket ,So Done print only once, outside the loop

Corrected 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");
                }
        }
}
```

Snippet 7:

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

Error:

```
UninitializedWhileLoop.java:4: error: variable count might not have been initialized
while (count < 10) {
       ^
1 error
```

Count variable is not initialized with value.

Corrected 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);
}
}
```

The code only print 1 because num is initialized to 1 and then decrement it and while loop only check num > 0.

Corrected code:

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

it prints 0 2 4 , but if we want to print from 0 to 4 use i++ instead of i+=2 .

Corrected 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:

```
IncorrectWhileLoopControl.java:4: error: incompatible types: int cannot be converted to boolean
while (num = 10) {
           ^
1 error
```

In while condition, Assignment (=) is used, to run the code properly I have given num > 10.

Corrected code:

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

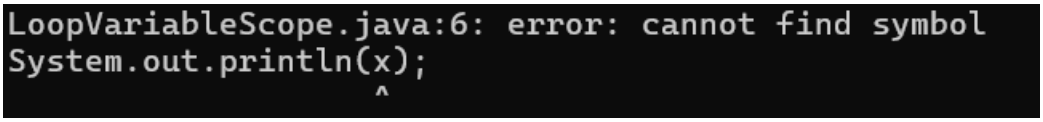
The output is 0 2 4.
To print from 0 to 4 update i+=2 to i++ so that 'i' variable will increment by 1 only not by 2.

Correceted 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
    }
}
```



Variable x is initialized inside a block and it is only accessible by that block only not outside the block.

Corrected code:

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