

COMP 110/L
Fall 2022

Lecture 11 Handout

1.) Consider the following code:

```
Scanner input = new Scanner(System.in);
int x = input.nextInt();
int y = input.nextInt();

if (...) {
    System.out.println("Condition was true");
} else {
    System.out.println("Condition was false");
}
```

... needs to be filled in with a boolean expression above.

1.a.) Write a condition which will be `true` if `x` is greater than `y`, **and** `y` is **not** equal to 7.

1.b.) Write a condition which will be `true` if `x` is less than `y`, **or** `x` is equal to 8.

1.c.) Write a condition which will be `true` if `x` is in the range 0 to 10, inclusive for both 0 and 10.

1.d.) Write a condition which will be `true` if `x` is **outside** the range 0 to 10. Use boolean negation to invert the result from 1.c.

2.) Consider the following class definition:

```
public class MyClass {
    public static int foo(int input) {
        if (input < 10 || input == 17) {
            return 0;
        } else if (input > 20 && input < 40) {
            return 1;
        } else if (!(input == 16)) {
            return 2;
        } else {
            return 3;
        }
    }
}
```

Write JUnit tests that will test each possible output of the `foo` method. You'll need at least 4 tests in total. You may wish to write additional tests that test alternative paths through the code (e.g., there is more than one way to return 0). The first test is partially provided.

```
import static org.junit.Assert.assertEquals;
import org.junit.Test;

public class MyClassTest {
    @Test
    public void testLessThan10() {
        assertEquals(0,
```