Solutions to XInteract problems:

• [Box Test.] The trick is to realize that if the upper left corner is at \((x, y)\) then starting with \(x\) and counting a total of \(w\) pixels, any pixel on the right edge has x-coordinate \(x + w - 1\). A similar comment holds for \(y\) and \(h\). In other words, \(x \leq x_1 < x + w\) and \(y \leq y_1 < y + h - 1\).

\[
x_1 \geq x \land y_1 \geq y \land x_1 < x + w \land y_1 < y + h
\]

• [Compound “if” condition.] The trick is to realize that “between” could mean either than \(a\) is to the right of \(b\) and to the left of \(c\), or that \(a\) is to the right of \(c\) and to the left of \(b\).

The long answer is:

\[
\text{if } ((a \geq b \land a \leq c) \lor (a \geq c \land a \leq b)) \{ \\
\text{ } k = \text{true}; \\
\text{ } \}
\]

\[
\text{else } \{ \\
\text{ } k = \text{false}; \\
\text{ } \}
\]

The more elegant (and shorter) answer doesn’t even use an “if”:

\[
k = (a \geq b \land a \leq c) \lor (a \geq c \land a \leq b);
\]

• [Compound Conditions.]

\[
i > 0 \land i < 10
\]

• [Find Minimum Input.] The Scanner variable is named input, not _input — some people were confused by this inconsistency with the example done in class.

\[
\text{minimum = Double.POSITIVE_INFINITY;} \\
\text{while (input.hasNext())} \{ \\
\text{ } double value = input.nextDouble(); \\
\text{ } \text{if (value < minimum) } \{ \\
\text{ } \text{minimum = value;} \\
\text{ } \} \\
\text{}
\]
• [If: Odd or Even.]

    if (j % 2 == 0) {
        k = 0;
    }
    else {
        k = 1;
    }

    NOTE: A much shorter answer is simply:

    k = j % 2;

• [Inheritance: Constructor for a subclass (fixed).] The first thing the Scholar constructor must do is call the constructor for the superclass. Since this constructor requires a string for the name, we must supply it with one.

    public Scholar(String name, int num) {
        super(name); // NOTE: super(); won’t work
        _numberOfAwards = num;
    }

• [Loop: Count Backwards by .5.]

    double value = 2.0;
    while (value >= -2) {
        System.out.println(value);
        value = value - .5;
    }

• [Loops, methods: exponents.]

    public int exponent(int x, int y) {
        // Special case: y == 0
        if (y == 0) {
            return 1;
        }
        int power = 1;
        while (y > 0) {
            power = power * x;
            --y;
        }
        return power;
    }

• [Method: Ascending Order (fixed)]
```java
public boolean ascend(int a, int b, int c) {
    return a <= b && b <= c;
}

• [Method: Inches to Centimeters]

    public double convert(int inches) {
        return inches * 2.54;
    }

• [Sum Positive Inputs.]

    while(keyboard.hasNext()) {
        double value = keyboard.nextDouble();
        if (value <= 0) {
            break;
        }
    }

• [Sum to Limit.]

    _sum = 0;
    int nextValue = 1;
    while (_sum < n) {
        _sum = _sum + nextValue;
        ++ nextValue;
    }

• [Sum to limit(2).]

    _sum = 0;
    int nextValue = 1;
    while (_sum + nextValue <= n) {
        _sum = _sum + nextValue;
        ++nextValue;
    }

• [SumLoop.]

    _sum = 0;
    int count = 1;
    while (count <= n) {
        _sum = _sum + count*count;
        ++ count;
    }
```

Handed out on 4 April 2007  Handout # 17