**CS201 Lab 3**

**35 points**  **Due**: 02/9/15

**Problem:** It’s February, which means time for winter sports! One Olympic sport is the ski jump, where the score is determined by distance traveled after skiing down a ramp. What type of speed does a ski jumper need to achieve on the ramp to make a good distance on their jump? Let’s make a program to calculate it and determine how many points they’d receive if they went that distance.

**Purpose:** This lab gives you practice with:

* Using JOptionPane for input and output
* Using the Math class (<http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html>, textbook)
* Using decision making in your code (if/else-if/else)
* Using System.exit(1) to end a program that cannot continue due to error
* Drawing a flowchart & finding control paths

**Details:**

Given the type of ski jump (normal vs. large) and the jumper’s speed at the end of the ramp, predict how far they will jump using the simplification shown below. After determining their distance, you can also calculate how many points they would get on that hill. For a normal hill, 90m is par (i.e. the average expected result) and on a large hill 120m is par. If they are below par, they lose points for each meter; if they are above par, they gain the same amount of points per meter.

The algorithm below basically follows these steps: 1. Declare variables, 2. Ask user for necessary information, 3. Set variables based on user input, 4. Calculate distance and points, 5. Output results.

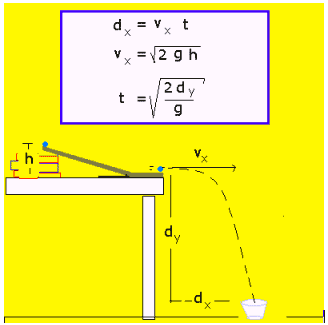
*Write your code to follow this algorithm:*

0. Declare height (short), points\_per\_meter (double), par (int), velocity (double), hill (String)

1. Ask the user for the jumper’s velocity at the end of the take-off ramp, as a real number in m/s

2. Ask the user if the jump is on the normal ramp or the large ramp; ask them to enter 1 for normal, or 2 for large.

3. If the jump is on the normal ramp, then

a. set height to 46

b. set points\_per\_meter to 2

c. set par to 90

4. Otherwise, if the jump is on the large ramp, then

a. set the height to 70

b. set points\_per\_meter to 1.8

c. set par to 120

5. Otherwise:

a. output “error, not valid input”

b. end program using System.exit(1);

6. Calculate the time in the air as

7. Calculate the distance as velocity \* time

8. Calculate the points earned as 60 + (distance - par)\*points\_per\_meter

9. Output the predicted time in the air and distance traveled

10. If the points earned is greater than 60, output “Great job going so far!”

11. Otherwise, output “Better luck next time!”

**Steps:**

1. Create a new Java file and save it to your Lab3 repository folder.
2. Write your Java code following the above algorithm. **Use JOptionPane for all input and output.**
3. Write comments in your code to make it clear what it is doing.
4. Draw a flowchart of your code
5. Label the control paths in your flowchart
6. List the test cases based on your control paths.
7. Use the test cases to make sure your program works correctly, and fix it if it doesn’t. Don’t just assume you did it right, there are many things that could have gone wrong.
8. Include an updated version of the header comments. Many lines should change!

**Make your output easy to understand.**

**Submit:**

1. To GitHub:
   * Your .java file
   * Your test cases, labeled by control path
2. On paper in class:
   * A short reflection about what you learned in lab, what it was like working with your partner, and what gave you the most trouble. (1 per person; .doc, .docx, or .pdf)
   * A hardcopy of your flowchart with control paths labeled (1 copy per pair)