**CS 203 HW #4 Golf Simulation
Spring 2012**

**Due Dates:** Part A: Wednesday, Feb. 15 at 11 AM (code only)
                   Part B: Wednesday, Feb. 22 at 11 AM (code and report)

You will complete a Java graphics program that draws the path of a golf ball. This assignment will give you practice with loops (while and for loops), conditionals, graphics, and using the Math class.

**Specification (Part A):**

1. The program should open a dialog box to prompt the user for the desired time increment (in seconds) to use in the visualization. The program should continue to re-prompt the user when the user enters a value <= 0. [Note: your program does not need to handle non-numerical input typed by the user.] **\*\* This has been done for you in the starter code \*\***
2. (1 pt) The program should open a dialog box to prompt the user for the desired velocity (in m/s) of the golf ball. The program should continue to re-prompt the user while the user enters a value <= 0. Again, you may assume the user types in numbers.
3. (1 pt) The program should open a dialog box to prompt the user for the desired angle (in **radians**). The program should continue to re-prompt the user while the user enters a value <= 0 or >= PI/2. You may assume the user types in numbers.
4. (1 pt) The program should open a dialog box to prompt the user for the desired number of clouds (0 to 20, inclusive). The program should continue to re-prompt the user while the user enters an invalid integer. You may assume the user types in numbers.
5. (4 pts) The program should draw the following objects in the paint method. You get to choose their locations on the screen (use reasonable locations).
	1. The sky background
	2. The fairway for the hole
	3. The green (oval) [You may determine how far the green will be from the starting location of the ball]
	4. The golf ball
	5. The pin and flagstick
	6. A scoreboard
	7. One cloud (at a random location on the screen)

**Submit Part A (just the code) to Moodle. Part A is worth 7 points toward a total of 27 points on this assignment. Part A will be graded on functionality only.**

**Specification (Part B):**

1. (4 pts) The program should draw the position of the golf ball at each time increment while the golf ball is in the air (y position >= 0). You may assume no wind resistance. One pixel will represent one meter to keep things simple. Here is the algorithm:
	1. Create and assign double variables to represent the x and y positions. Assign xPosition and yPosition to 0.
	2. Create and assign a double variable for time to 0.
	3. While yPosition >= 0.0:
		1. Calculate the x at time: xPosition = velocity \* Math.cos(angle) \* time
		2. Calculate the y at time: yPosition = velocity \* Math.sin(angle) \* time - .5 \* 9.8 \* time \* time
		3. Update the time by adding the user-specified timeInterval to it.
		4. Draw the golf ball (remember – the graphics window is positive y going down, so you will need to translate the physical (x,y) coordinates of the golf ball to graphical coordinates. This will likely involve doing something like yDraw = 500 – yPosition.) You may draw the golf ball much larger than actual scale (about 3 pixels in diameter), so it can be seen on the screen.
2. (1 pt) The program should display on the scoreboard three items (described below).
	1. Calculate the total distance the ball flies in the x direction: velocity\*velocity\*Math.sin(2\*angle)/9.8
	2. Using this distance, determine how far away the ball landed from the hole.
	3. Display on the scoreboard how far (in total meters) the ball flew.
	4. Display on the scoreboard how far the ball landed in meters from the hole. Math.abs might be handy here.
	5. Determine if the ball landed on the green (no bounces) and display “Hit the green” or “Missed” on the sign.
3. (2 pts) The program should draw the user-specified number of clouds in the sky. Tammy drew clouds as overlapping ovals, but you can use fancier shapes if you would like. Put these in random locations (see the starter code for creating a random number). [clouds should be drawn inside a for loop]

**Submit Part B (code and report) to Moodle. Part B is worth 20 points toward a total of 27 points on this assignment and will be graded on functionality, quality, and the summary report.

Helpful Hints:**

* To use sin and cos functions, the syntax is Math.cos(angle) and Math.sin(angle)
* Your distance calculations will most likely be using doubles, but you will need to convert these to ints to use as coordinates for drawing. You can cast a double to an int with (int).
* Be sure to translate "physical" coordinates in 2D space to Java's window coordinates. Remember that x goes from 0 to 500 (left to right), but y goes from 0 to 500 (top to bottom).
* If you need to access variables from the Golf() constructor and paint method, you must declare them as instance variables just under the class definition. (See starter code.)
* Put all your dialog windows in the Golf() constructor and all your drawing in the paint method.

**Additional Enrichment:**

If you have time, consider adding extra features to your program (and document these features in your summary report):

* Add other interesting parts to the golf hole. For example, you might add sand traps, water hazards, and trees. Then, the sign could display a message such as “Landed in the water” or “Landed in the bunker” instead of “Missed the green”.
* Add the physics for wind resistance.
* Draw the ball bouncing after landing on the ground.
* Determine the maximum height of the golf ball.
* Add seats and fans to the background.
* Let the user define how far the green is away from the start location of the golf ball and how large the green is. Draw the green with the user-defined dimensions and location.

**Logistics:**

1. Download the starter BlueJ project. Remember to include comments in your code to remind the reader and yourself about the functionality of different parts of your program. Please keep the name of the java file Golf.java.
2. For Part A, be sure your name is in the comments and upload just the Golf.java file to Moodle.
3. For Part B, create a zip file containing your BlueJ files and your report. Name your zip file username\_HW4.zip and submit to Moodle.

**Grading Guidelines:**Part A will be graded on a scale of 0 to 7 based on functionality. Part B will be graded on a scale of 0 to 7 in two categories:

* Code Quality: Design, Documentation, Readability, Good names, Whitespace, Indentation
* Code Operation: Functionality of code and adherence to project specification

Your summary report will be graded on a scale of 0 to 6 based on:

* Correct use of technical vocabulary
* Clarity

**Report Guidelines and Format: (use the template for HW1 but answer these questions instead)
1. Screen Shot:** Put a picture of your program here. You can use the Print Screen button to get a copy of your screen and then paste this into MS Word.

**2. System Description:** 1. Describe how the loop draws the ball motion. 2. Describe how the loop draws clouds.

**3. Testing and Evaluation:**

**3.1 Meets Specification** If your program does not meet the specifications, please note these differences. If it includes more than the specification, note these additions. If it meets the specification, just write “This program meets all specifications.”

**3.2 Test Chart** Include a table containing a test chart that shows the input values you tried and the results (landed on green or missed and number of clouds). Your table should include columns for each input and each output and list all the combinations you tried. For each test case, label it as T (typical), E (edge case), or I (invalid case). See below for an example.

 **Input** **Output**

TimeInc Vel Angle NumClouds | Hit/Missed NumClouds

.2 (T) 80.2 (T) .8 (T) 15 (T) Missed 15

….

**3.3 User Test** Have a friend who is not enrolled in CS203 run your program. What did he/she like? What could be improved? Did he/she get the ball to land on the green?

**4. Conclusion:** Describe the most challenging aspect(s) of this program and what you learned from completing it.

1. How long did you spend on this assignment (part A plus part B plus the report)?

2. By typing your name here, you are acknowledging that the code and report you are submitting are your own.

**Appendix:** Copy and paste your code here (use **Courier** font so the characters line up correctly).

Please keep your reports concise (no more than 2 pages - not including your test cases and test chart). A high quality computer scientist can explain their code, their process, and their testing to their colleagues and bosses. When writing the report, your audience is someone else taking CS 203. I ask you to do these reports because you will be expected to explain your ideas, whether or not you become a computer scientist.