Assignment #2
Conditional & Repetitive Statements
Due: Monday, 01/28/13, 11:59pm

(75 pts) Problem Statement: Write two C++ programs to practice conditional and repetitive logic. In addition, you will read input from the user.

- Write a C++ program, buoyancy.cpp, to read the weight, (in pounds) and radius, r, (in feet) of a sphere as input from the user using the C++ cin input stream, and output whether the sphere will sink or float in the water. Use $\gamma = 62.4 \text{ lb/ft}^3$ as the specific weight of water, and the volume, $V$, of the sphere is $(4/3)\pi r^3$. The buoyant force can be computed by

$$F_b = V \times \gamma$$

where $F_b$ is the buoyant force, $V$ is the volume of the submerged object, and $\gamma$ is the specific weight of the fluid. If $F_b$ is greater than or equal to the weight of the object enter by the user, then your program will output, “This sphere will float”, otherwise your program will output, “This sphere will sink.”

Then, ask the user if he/she wants to find out whether another sphere will sink or float based on a different weight and/or radius. If the user wants to calculate buoyancy again, then repeat the process above, asking the user for the weight and radius of the sphere, calculating the buoyant force, outputting a message, etc.

**Revised from Savitch, Programming Projects #6, p. 95**

- Write a C++ program, primes.cpp, which finds and prints all of the prime numbers between 3 and some upper limit, $x$, input by the user, e.g. 100. A prime number is a number that can only be divided by one and itself, i.e. 3, 5, 7, 11, 13, 17, ...

One way to solve this problem is to use a doubly-nested loop. The other loop can iterate from 3 to $x$, while the inner loop checks to see whether the counter value for the outer loop is prime. One way to decide whether the number $n$ is prime is to loop from 2 to $n-1$; if any of these numbers evenly divides $n$, then $n$ cannot be prime. If none of the values from 2 to $n-1$ evenly divide $n$, then $n$ must be prime.

Your program must print all the prime numbers between 3 and $x$. What happens when $x$ becomes large, e.g. 1,000 and 1,000,000?

**Revised from Savitch, Programming Projects #4, p. 94**
(10 pts) In your implementations, make sure that you include a program header/description in your program, in addition to proper indentation/spacing and other comments! Read the class style guideline for more information: http://classes.engr.oregonstate.edu/eecs/winter2013/cs161-001/161_style_guideline.pdf

You are graded on having a header, proper comments, and readable code with indentation and vertical spacing that is CONSISTENT throughout your program. DO NOT align your entire program on the left side. This will cause you to automatically lose the full 10 points. In addition, do not forget your program header!!!

(15 pts) You are required to turn in a written document (as a pdf) addressing Polya’s steps to solving a problem for both problems above, with step 3 being the C++ programs you write to carry out/implement your plans. With this said, your written document must include these three sections:

**Understanding the Problem**
In your own words, explain what YOU think the problem is asking you to do. In this section, document your uncertainties about the problem and anything else that you feel was unclear or vague. This is to ensure that YOUR understanding matches MY understanding of the problem.

**Devising a Plan/Design**
At a minimum, provide an algorithm/pseudo code you designed to help solve the problem. In addition, include pictures/flow charts you used to help you devise your plan, as well as any other design decisions you made such as how to manage your time, how to decompose the problem, where to start first, etc. You can scan any handwritten work and attach it to the document as needed.

**Looking Back/Self-Reflection**
Report any checking/self-reflection you did while solving the problem. For instance, how did you make sense of the output from the implementation? This includes things such as using a calculator to make sure the output is correct, testing to make sure your code executes correctly and behaves the way you expect under specific circumstances, using external sources of information such as the internet to make sense of the results, etc. Also, include a statement about what you learned from the assignment.

Electronically submit your two C++, buoyancy.cpp and primes.cpp, program files and your design document, .pdf, by the assignment due date, using TEACH.