Assignment #2
Variables and Conditionals
Due: Sunday, 10/08/17, 11:59pm

Grading: Every assignments in this course is graded by demoing your work for 10 minutes with a TA. You are required to **meet with a TA within two weeks of the due date** to demo. You can schedule a demo with a TA on the **home page** in the far right column of the bottom table labeled "Grading Hours".

- **Demo Outside 2 Weeks:** Assignments that are not demo'd within the acceptable time period will be subject to a 50 point deduction.
- **Demo Late Assignments:** Late assignments must still be demoed within the two week demo period beginning from the assignment's original due date.
- **Missing a Demo:** If you miss your demo with a TA, you will receive a 10 point (one letter grade) deduction to that assignment for each demo missed.

Your job is to convince the TA that your program works correctly, i.e. show your TA how to use/break your program☺

Design is very important when developing programs and there are a variety of ways to approach it. You may draw pictures, write it out in prose or structured text, use pseudo code, and more! The point of design is to give you a blueprint to follow while you are coding. This saves time debugging your program as you can catch mistakes early. It is better to spend one hour of designing than it is to spend five hours debugging.

For this assignment **you must design a solution** to the following problem statement, and **implement your solution**!

Example Design Document: **Polya_template.pdf**

- Understanding the problem. (*Recognizing what is asked*.) **(5 pts)**
- Devising a plan. (*Responding to what is asked*.) **(10 pts)**
- Carrying out the plan. (*Developing the result of the response*.) **(75 pts)**
- Test Plan/Looking back. (*Checking. What does the result tell me?*) **(10 pts)**

Problem Statement

The local middle school would like some text adventure games to keep their students occupied during down time. The school is leaving it up to your skill and good judgement to develop a game. It is up to you what the story and theme is but there are some requirements:

- There must be **at least five different paths or solutions** to complete the adventure
- There must be **an element of chance** that would change a user's chosen path
- You must **handle invalid input** from the user.

An example of the run of the program looks like this:

Hello and welcome to the adventure!
To go right enter 1, to go left enter 2:
1

You chose to go right. You have now entered Scandinavia and are being hunted by friendly oxen.
Enter 1 to befriend the oxen, enter 2 to run from the oxen: 1

You attempted to befriend the oxen. You think you can ride
them. Enter 1 to attempt to ride the oxen, enter 2 to walk away:
1

Unfortunately fate was not on your side (due to a random number, coin flip, etc.), and you are forced to walk away.
Enter 1 to walk right, enter 2 to walk left: 1

You chose to go right. You have now entered America and are fighting a strong warrior. Enter 1 to throw your arrow at the warrior, enter 2 to run home: 2

You chose to go home. The adventure has ended.

The rest of the implementation is up to you, but try to make your game as clean and attractive to play as possible.

(25 pts) **Design**
Understanding the Problem (5 pts)
What is the problem asking you to do? What are the inputs for the problem?
What questions do you have about the problem?

Devising a Plan (10 pts)

- What is the theme of the adventure going to be?
- What are the options?
- List the decisions you will give in your text adventure?
- Which decisions will have an element of chance?
- How are you going to create the element of chance?
- How are you going to handle invalid input entered by the user?

Test Plan (10 pts)
You need to include good and bad values for your program.

Input Values:	Expected Outcome:	Actual Outcome:
100	Error Message	Crashed
1	Takes me to the right to Scandinavia with friendly oxen.	Takes me to the right to Scandinavia with friendly oxen.
…	…	…

(65 pts) **Implementation**

In addition to design, you must implement your program as a C++ program. Here are implementation requirements:

*There must be an empty line separating each navigation
*The choice system must be based on numbers
*You must handle invalid input from the user that is not one of the allowed choices.
*You must use if/else statements and/or switch statements
*You need to use the rand() function to add an element of chance
*You must have 5 different paths/solutions.

(10 pts) **Extra Credit**
Continue to ask the user if he/she wants to play the adventure game again until the user decides they do not.

(10 pts) **Program Style/Comments**

In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Below is an example header to include. Make sure you review the style guidelines for this class, and begin trying to follow them, i.e. don't align everything on the left or put everything on one line!
http://classes.engr.oregonstate.edu/eecs/fall2017/cs161-001/assignments/161_style_guideline.pdf

```
/*************************************************
** Program: adventure.cpp
** Author: Your Name
** Date: 10/06/2017
** Description:
** Input:
** Output:
*************************************************/
```

Electronically submit your C++ program (**.cpp** file, not your executable!!!) by the assignment due date, using TEACH.
https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth

Make sure you demo Assignment #2 within two weeks of the due date to receive full credit. If you go outside the two week limit without permission, you will lose 50 points. If you fail to show up for your demo without informing anyone, then you will automatically lose 10 points.