# University of Pittsburgh
# Department of Electrical and Computer Engineering ECE 0301 – Spring 2021

## In-Class Assignment #2

Programming concepts:

- C++ string variables
- Boolean variables and logical operators

ECE concepts:

- Propositional logic and evaluating Boolean expressions

Background: The following are propositions, i.e. statements that are objectively true or false:

$P_1$ Pigs can fly.
$P_2$ Earth is flat.
$P_3$ The moon is cheese.

We will refer to these as *elementary propositions*, because they cannot be divided into phrases or clauses, each of which is also a proposition. By contrast, a *compound proposition* is composed of two or more propositions, such as:

Pigs can fly, and the moon is not cheese.

Clearly, the truth value of a compound proposition depends on the truth values of the elementary propositions from which it is formed. In this assignment, you will write a program that evaluates and displays the truth values for 8 compound propositions, given the truth values of the elementary propositions.

Instructions:

Develop software according to the following specifications and submit whatever portion you have completed to the class repository before midnight tonight. **For all in-class assignments, you must include enough comments to explain what your code does and why! You must also choose meaningful names for all variables so that it will be easy for a reader to understand your code.**

1. Write a computer program that will display the following introductory message to standard output:

```
ECE 0301 Propositional Logic Example
```

**Don't forget to include comments! Type them in your code as you complete each item.**

2. Declare three variables from the C++ `string` class, and initialize them with the three elementary propositions $P_1$, $P_2$, and $P_3$ described in the Background section (don't forget the `#include` directive to use the `string` class).

   Write three lines of text to standard output to report the propositions, e.g.

   ```
   P1: Pigs can fly.
   ```

3. Declare three variables of type `bool` to store the truth values of the propositions. Prompt the user to enter the truth value of $P_1$, e.g.

   ```
   Enter P1 truth value:
   ```

   Read the user response from standard input; the user will type 0 for false and 1 for true.

   Repeat for propositions $P_2$ and $P_3$. The user must enter the truth values in numerical order.

4. Write one line of text to standard output, to display what the user has entered, e.g.

   ```
   You entered: P1 = true, P2 = false, P3 = false.
   ```

   Use the `boolalpha` manipulator to set the `boolalpha` flag for the `cout` stream, so that the words `true` and `false` are displayed, instead of the numeric values `1` and `0`. The `boolalpha` manipulator is not covered in the Gaddis text, but the syntax is similar to the other stream manipulators described in Chapter 3, for example:

   ```
   bool x = true;
   cout << “Boolean variable x is “ << boolalpha << x << endl;
   ```

5. Consider the following compound proposition,

   $P_4$: Pigs can fly, Earth is not flat, and the moon is cheese.

   Declare an additional `bool` variable to store the truth value of $P_4$, and use logical operations to compute the truth value from the truth values of $\{P_1, P_2, P_3\}$.

   Write seven lines of text to standard output to report the truth values of $\{P_1, P_2, P_3, P_4\}$ in a table, using the following format,

   ```
   Truth Table:
   Proposition    Truth Value
   -----------    -----------
   P1             true
   P2             true
   P3             false
   P4             false
   ```

   Use the tab character `\t` to align the columns of your table. You may have to experiment with the number of tabs to include on each row of output so that the columns are aligned.

   Run your code, and ensure proper execution, and that the columns of your table are properly aligned. Run your program 8 times, changing the truth values of $\{P_1, P_2, P_3\}$ each time so that all 8 combinations are tested, and ensure that the column alignment is correct each time, and that the truth values of $P_1$ - $P_4$ are correct each time.

6. Consider the following compound proposition,
   $P_5$: Pigs cannot fly, or Earth is flat, or the moon is cheese.

   Declare an additional `bool` variable to store the truth value of $P_5$, and use logical operations to compute the truth value from the truth values of $\{P_1, P_2, P_3\}$.

   Add a line of output to your table to display the truth value for $P_5$, following the same format.

   Run your program 8 times, changing the values of the `bool` variables each time so that all 8 combinations are tested, and ensure that the column alignment is correct each time, and that the truth value of $P_5$ is correct each time.

   When you are certain your program is correct, save it in a file named:
   `ece0301_ICA02_step06.cpp`
   Submit this file to the class repository.

7. Consider the following compound proposition,
   $P_6$: One of the elementary propositions $\{P_1, P_2, P_3\}$ is true, and the other two are false.

   Declare an additional `bool` variable to store the truth value of $P_6$, and use logical or arithmetic operations to compute the truth value from the truth values of $\{P_1, P_2, P_3\}$.

   Add a line of output to your table to display the truth value for $P_6$, following the same format. Test your program as described in step 6. It is **not** necessary to submit your code to the class repository for this step.

8. Repeat step 7 for the compound proposition,
   $P_7$: None of the elementary propositions $\{P_1, P_2, P_3\}$ are true.

9. Repeat step 7 for the compound proposition,
   $P_8$: A majority of the elementary propositions $\{P_1, P_2, P_3\}$ are true.

10. Repeat step 7 for the compound proposition,
    $P_9$: Pigs cannot fly, and either Earth is flat or the moon is cheese, but not both.

    Test your program fully. It should display the truth table for all 9 propositions. When you are certain your program is correct, save it in a file named:
    `ece0301_ICA02_step10.cpp`
    Submit this file to the class repository.

**Don't forget to include comments! You will lose credit if you leave them out, even if your code functions as directed!**