ICS4U **Assignment – Pascal's Triangle** Mar 10, 2010

Pascal's triangle is a triangular array of numbers named in honour of the French mathematician Blaise Pascal. The first few rows of the triangle are:

```
                    1
               1         1
          1         2         1
     1         3         3         1
1         4         6         4         1
```

The triangle can be extended indefinitely.

Each number in Pascal's triangle is one of the *combinations* C(n, k), which is mathematically defined as:

$$C(n,k)=\frac{n!}{k!(n-k)!}$$

In terms of *combinations*, Pascal's triangle can be expressed as:

```
                        C(0,0)
                C(1,0)          C(1,1)
        C(2,0)          C(2,1)          C(2,2)
C(3,0)          C(3,1)          C(3,2)          C(3,3)
C(4,0)          C(4,1)          C(4,2)          C(4,3)          C(4,4)
```

The *combination* C(n, k) can be written in terms of the *permutation* P(n, k), where:

$$P(n,k)=\frac{n!}{(n-k)!}$$

So now we can write $C(n,k)=\frac{P(n,k)}{k!}$

The symbol $n!$ is read as “n factorial”. The value of $n!$ can be found using the formula:

$$n!=\begin{cases}1 & \text{if } n=0\\ n\times(n-1)\times(n-2)\times\ldots\times2\times1 & \text{if } n>0\end{cases}$$

Problem Statement

Design and write a program that prints portions of Pascal's triangle. The program should repeatedly ask the user to supply the number of rows in the triangle. The program should terminate when the user supplies the value zero.

In printing a triangle, the first value in the last (bottom) row should appear along the left edge of the display (i.e., the first column of the display). Each preceding row should start printing 3 spaces further to the right.

The number of spaces between entries in a row should be adjusted so that the values are aligned correctly from row to row (see the example at the beginning of the assignment).

Your program should be able to print up to about 20 rows of Pascal's triangle. As the number of rows gets larger, there will be issues with the width of the display. If the user requests a number of rows greater than what will display nicely, the program should reject the value with a warning.

Notes:

1. You must submit a written or typed design (pseudocode or flowchart) *before* you begin coding. If you cannot design a solution, then you cannot properly code a solution.

   It is recommended that your design use an *iterative* or *stepwise refinement* process, as you have been shown in previous examples. Start with something general and refine your design by asking questions and addressing specific “how to” algorithms.

2. This assignment focuses on *methods/procedures/functions*, and as such, you must make heavy use of them to solve your problem. It is recommended that you isolate the functionality of the following using *methods/procedures/functions*:
   (a) factorial
   (b) permutation
   (c) combination
   (d) user input
   (e) printing rows

3. Documentation is required. Your program design will address part of this requirement. Your source code must also be fully commented.

   Some advice on good comments:

   (a) Use meaningful names for all identifiers unless their purpose is *painfully* obvious.
   (b) Align/indent comments with the code block they describe.
   (c) Comment while you code. Commenting after the code is complete takes more time, and you will usually over-comment. Plus you might miss something important.