# CIT-237
# Command-Line Arguments

December 9, 2019

# Reminder / Announcement

- Quiz 8 will be on Wednesday, December 11.
- The material covered on Quiz 8 will be:
  - The Lectures of November 25, 27, and December 2.
  - Chapters 19, 20 and 21.
- The EXTENDED Due date for Project 3 is TODAY (Monday, December 9).
- Our last day of class is Monday, December 16.
  - Obviously, that is the last day to demonstrate the solutions to Lab Exercises.
  - We will NOT have a Final Exam.

# Command-line Environment (1)

- While we are accustomed to using our computers with the "point-and-click" graphical user interface, there are some occasions when a program needs to be invoked from a "command-line" environment.
- The Windows Operating System allows us to open a command-line interface by running a special tool:
  - In Windows 7, the "default" command-line interface was the "cmd" tool.
  - In Windows 10, the "default" command-line interface is the "Power Shell" tool.
- There is a rich set of command-line operations that the operating system supports.
  - To view a summary of these operations, open a "cmd" tool and type "help", followed by the ENTER key:

```
help
```

  - If you wish to save a copy of this "help" information for browsing later, type the following (followed by the ENTER key):

```
help > help.txt
```

# Command-line Environment (2)

- To view more detailed documentation about a particular command, type “help”, followed by the command name, followed by the ENTER key.
  - Several examples:

```
help dir
help cd
help set
```

- This detailed documentation can also be saved in a text file, if desired. For example:

```
help dir > dir.txt
help cd > cd.txt
help set > set.txt
```

# Command-line Environment (3)

- Many details of how the system behaves can be altered by modifying *environment variables*.
  - This is described in the documentation for the "set" command.
  - To view the **current** value of all environment variables, run the "set" command with no arguments:

```
set > envirVars.txt
```

- There is a convenient way to start the command-line interface at a particular location :
  1. Use the *File Explorer* tool to navigate to a particular folder.
  2. Point the mouse at an "empty" column of the display,
  3. Hold down the SHIFT key,
  4. Press the right mouse button, and
     - For Windows 7, select the "**Open Command Window Here**" menu.
     - For Windows 10, select the "**Open Power Shell Window Here**" menu.

# Windows 10 Command-Line Environment

- As previously noted, the Operating System I currently run at home is Windows 7.

  (I have deferred upgrading until this semester is over.)
- Most students today run Windows 10 on their personal computers.
- There are numerous online tutorials about the Windows 10 command-line environment.

# C++ Command-line Arguments

- In a C++ program, command-line arguments may be passed to the "main" function in the form of optional calling parameters:

```
int main(int argc, char *argv[])
```

- The **`argc`** parameter contains the number of items that were typed on the command line, including the name of the program.
- The `argv` parameter is an array of char pointers.
  - In the function header, the brackets are empty because `argv` is an external array of unknown size.
  - The number that is stored in `argc` , however, is the number of elements in the `argv` array.
  - Each pointer in the `argv` array points to a C-string holding one command line argument.

# Command-line Arguments Example

```
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
   cout << "You entered " << (argc - 1);
   cout << " command line arguments.\n";
   if (argc > 1)
   {
      cout << "Here they are:\n";
      for (int count = 1; count < argc; count++)
         cout << argv[count] << endl;
   }
   return 0;
}
```

# Adding a List of Numbers (1)

- Another example is a program to add a list of numbers, where the numbers are listed on the command line.
- One nice feature of this is that the program will add as many numbers as the user chooses to type on the command line.

# Adding a List of Numbers (2)

```
#include <iostream>
#include <cmath>  // Needed for atof
using namespace std;

int main(int argc, char *argv[])
{
   double total = 0;

   if (argc > 1)
   {
      for (int count = 1; count < argc; count++){
         total += atof(argv[count]);
      }
      cout << total << endl;
   }
   return 0;
}
```

# Appendix Document on Moodle

As I indicated in the previous class, I have been unable to locate the Appendices for the 9th edition of the textbook on the publisher's website.

- The document I have placed on Moodle is from the 8th edition.