# Visual Studio and getopt

One problem with using Visual Studio with EECS 281 is that the IDE/compiler does not have the file getopt.h in its include folder, nor does a compiled version of the functions exist. You can overcome this by using two files supplied on CTools, getopt.h and getopt.c.

The getopt.c file is easier to use: simply put a copy in each project that needs it. When you turn in your project, do not make getopt.c part of your tarball.

The getopt.h file is a little more problematic. There are two approaches to solving this problem.

1) Easier but worse solution: Put getopt.h inside your project folder, and #include "getopt.h" instead of <getopt.h>. When you pack up your project to test on CAEN or submit, you have to change your include statement to <getopt.h>. You should also NOT include getopt.h as part of your submission tarball.

2) More difficult but better solution: Put the getopt.h file inside the Visual Studio include folder. For Visual Studio Pro 2015, this folder is usually located in:

c:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\include

If you're using Visual Studio 2013, the folder name is "Microsoft Visual Studio 12.0" instead of 14.0.

**Windows 7/8 method**

However this location is usually locked and can only be accessed if you are "Administrator". One way to do this is click Start -> Run, type in cmd, and instead of pressing Enter, press Shift-Control-Enter. You can then use the copy command to put the file in the correct location by hand, or type explorer to use the Windows Explorer to copy the file to the target folder.

**Windows 10 method**

Type Windows-X (using the Windows key like a shift key, hold it down and type X, then release the Windows key). From the menu that appears, choose "Command Prompt (Admin)". From here, you have to copy the file yourself using commands. Here's an example of two commands that worked for me, you will have to at least change the user name, if not the folder containing the getopt.h file:

```
cd "\Program Files (x86)\Microsoft Visual Studio 14.0\VC\include"
copy \Users\Paoletti\Downloads\getopt.h .
```

## Creating a Project

When you create a project, to be able to use the integrated symbolic debugger, you need to create the project correctly. Some settings will become the default (marked with a * below), others will needs to be changed every time (marked with ++).

When creating a new project, choose the type **Win32 Console Application ***
**UNCHECK** the box that says **Create directory for solution ***
Go to Application Settings (or Next) and select **Empty Project ++**

## Command-Line Parameters

After creating a project, you may want to pass command-line parameters when you run the program, such as flags, filenames, etc. To do this, right-click on the project name (in bold) inside of the Solution Explorer pane, choose Properties (the last option in the popup menu), then select Configuration Properties -> General -> Debugging. The second option on the right should be "Command Arguments"; left-click on the empty place to the right and type in whatever you need. For example, for Project 1 you could type in:

```
-q < test-1-qm.txt
```

Where test-1-qm.txt is the name of a text file inside of your project folder, that will be used as input. To create the input file, either use Notepad and Save As into the project folder, or use Windows Explorer to navigate to the project folder and right-click where there are no filenames to create a New -> Text Document. Put the input file(s) in the same folder as your .cpp files.