**GCC**

That's great! GCC (GNU Compiler Collection) is a powerful and widely used compiler for several programming languages, including C, C++, and Fortran. Learning GCC will enable you to compile and run programs written in these languages on various platforms.

To get started with GCC, here are the basic steps:

1. Install GCC: Depending on your operating system, the installation process may vary. For Linux, GCC is often pre-installed, but if it's not, you can use your package manager to install it. For Windows, you can install MinGW (Minimalist GNU for Windows) or Cygwin, both of which provide GCC for Windows environments. Mac users can install GCC via package managers like Homebrew or MacPorts.

2. Writing a program: Create a source file with the appropriate file extension for the programming language you are using. For example, use `.c` for C programs, `.cpp` for C++ programs, and `.f` or `.f90` for Fortran programs.

3. Editing the program: Use a text editor or an Integrated Development Environment (IDE) to write your code. Popular choices include Visual Studio Code, Sublime Text, Atom, and Code::Blocks. Save the file with an appropriate name and the correct file extension.

4. Compiling the program: Open a terminal or command prompt and navigate to the directory where your source file is located. Then, run the appropriate GCC command for your programming language and desired output. For example, to compile a C program named "myprogram.c" and produce an executable called "myprogram", you would use the command `gcc -o myprogram myprogram.c`. Similarly, you can use `g++` for C++ programs and `gfortran` for Fortran programs.

5. Running the program: After successful compilation, you can execute the generated executable. On Linux and macOS, use `./myprogram` in the terminal. On Windows, you might need to run `myprogram.exe`.

6. Learning advanced GCC features: Once you're familiar with the basics, you can explore more advanced features of GCC. This includes optimization flags, debugging options, linking external libraries, and other advanced compiler settings.

7. Referencing GCC documentation: The official GCC documentation is an excellent resource to deepen your understanding. It provides comprehensive information on compiler options, language features, and other related topics. You can find it at https://gcc.gnu.org/onlinedocs/.

Remember, learning GCC involves understanding the specific programming language you are using in conjunction with the compiler. So, it's essential to study C, C++, or Fortran alongside GCC to write effective programs.