Installing Haskell and VS Code for CS 1JC3

Dr. William M. Farmer, Curtis D'Alves, Jason Balaci

November 16, 2025

Haskell is a programming language specification for a pure, functional programming language. To write and execute "Haskell" programs, we need to install two things:

- A Compiler, so that we can convert Haskell programs into machine code and run it on our machines.
- An Integrated Development Environment (IDE), a source code aware text editor that adds features for building, running, testing, and analyzing software.

We will cover the installation of both of these components in this guide.

Compiler

For the course, we will use the Glasgow Haskell Compiler (GHC), an industry standard, actively developed Haskell compiler. We will additionally use Stack, a build tool that helps manage and organize Haskell-based software projects. Stack will be responsible for automating the process of compiling Haskell source code into executable programs via GHC, downloading external libraries (i.e., dependencies), and running our testing suites.

To install GHC and Stack, we will use GHCup, a tool for managing GHC and Stack versions installed on your machine. This is useful for developers working with multiple Haskell projects with different compiler requirements on the same system. The installation process is slightly different depending on your operating system. While the GHCup website will immediately detect your operating system and provide you with the corresponding installation instructions, we provide them here for your convenience. Please follow the instructions for your respective operating system.

Windows

For Windows users, there are two options: a native Windows installation (recommended) or emulated installation using Windows Subsystem for Linux (WSL) (a tool for running Linux on your Windows machine). Either way works well. However, if your machine has limited memory (i.e., 4GB RAM) or a weaker CPU, you may benefit from a native Windows installation. Finally, if you're experiencing significant issues with the native Windows installation, you may try using WSL instead.

Native Windows Installation (Recommended)

Please follow the official installation video (up to 1:34, ensuring that you press "Y" when prompted for installing stack). To confirm your installation succeeded, you should be able to open up a fresh Command Prompt window and run the following command, which should print out the version of stack installed:

```
stack --version
```

Windows Subsystem for Linux (WSL)-Based Installation (Alternative)

As a reminder, WSL is a tool for running Linux and Linux applications on your Windows machine. Unfortunately, the performance impact of running Linux on Windows is non-trivial, and you may experience slower performance than a native Windows installation. However, WSL is a great tool for learning Linux and Linux-based development, and it is widely used in industry.

If you already have WSL installed, you may jump to the installation instructions for macOS/Linux. If you do not already have WSL installed, please follow these steps to install it:

1. *Open PowerShell in Administrator mode*: Press the four-flagged Windows key, search for "PowerShell", right-click the PowerShell application option and click "Run as Administrator."

2. *Install WSL*: In the PowerShell window that appears, please type "`wsl --install`" and hit Enter. You will need to wait a few minutes for the installation to complete, and you might be presented with confirmation windows during the installation (please confirm them). Once it is complete, please…

3. *Restart your machine*. At this point, when you log in to your account, WSL's installation will be finalizing, and you will be presented with a terminal window that asks you to create a WSL user account and password. Please use a username and password that you will remember (e.g., your first name and something memorable).

4. At this point, using WSL, which is effectively a terminal window that opens up a Linux shell, you may follow the installation instructions for macOS/Linux.

macOS/Linux

Open up your terminal and run:

1. Open your terminal (macOS: CMD+SPACE > Search "terminal").

2. *Install GHCup*: Run the following command, using the recommended settings presented (just hit "Enter" for every option):

   ```
   curl --proto '=https' --tlsv1.2 -sSf https://get-ghcup.haskell.org | sh
   ```

3. Close and re-open your terminal. At this point, GHCup will be installed, and we can proceed to install Stack with it.

4. *Install Stack*: Run the following command:

```
ghcup install stack
```

5. *Confirm stack installation*: Run the following command, and ensure that a message along the lines of "Version ..., Git revision ...." appears:

   ```
   stack --version
   ```

At this point, you have GHCup and Stack installed, and you're ready to install your IDE!

Integrated Development Environment (IDE)

There are a few IDEs that you may use out there (e.g., emacs, neovim, etc.) using plugins. For the purposes of this course, we recommend Visual Studio Code (VS Code) with the Haskell language support plugin.

Installation

VS Code has an installer and its own plugin installation system that we will use to add Haskell support. Note: if you are a Windows user using WSL, please additionally install the WSL plugin for VS Code.

Please:

1. Download and run the VS Code installer respective of your operating system, using the recommended default installation settings.

2. Open VS Code.

3. On the left-side of the VS Code Window that appears, click on the Extensions icon (the icon with 4 squares), click on the search box that appears, search for "Haskell", and click "Install" for the option where the author is "Haskell" as well.

4. **WSL Users Only**: Please additionally install the WSL plugin.

At this point, you're ready to start developing Haskell software! You can follow the Getting Started video from 3 years ago to get started (note: start from 7:30, ignore the extra "Haskelly" plugin recommendation, and when running "`stack new MyProject --resolver=lts-XX.YY`", please use "lts-22.44" instead).

Have fun!