Lab 1 requires common C++ development tools and LLVM 15 tools. All requirements are packaged inside a docker image `lab1\_docker.tar.gz` and distributed to you. This document explains how to use this environment for doing lab 1.

## Instructions:

1. Install docker engine on your laptop if you haven't done so. See [here](https://docs.docker.com/engine/install/) for instructions
2. Download the two files we shared: lab1\_docker.tar.gz is a docker image for lab1, and lab1\_starter.tar.gz is the starter code for lab1.
3. Untar the starter code: tar -xzf lab1\_starter.tar.gz in a directory of your choice on your laptop
4. Install the docker image into your local registry
   1. Run docker load -i [PATH TO THE lab1\_docker.tar YOU DOWNLOADED]this step can take up to a couple minutes.
   2. Check that the image was added correctly. Run docker images and make sure see something with a tag compilers:0.2
5. Running the docker image
   1. Run docker run -it -v <**ABSOLUTE PATH** TO WHERE YOU UNTARED lab1\_starter.tar.gz>:/code compilers:0.2
   2. The option “-v <ABSOLUTE PATH TO WHERE YOU UNTARED lab1\_starter.tar.gz>:/code” simply map the directory containing the Lab1 starter code from your host machine into the docker container
   3. After the cmd in step a), you should already be inside a docker image. Navigate to /code/src (the path where the starter code is mapped inside the docker) and start lab1
   4. You can test that your environment is correct by navigating to /code/src and run make (this is also how you compiler your lab1 once you make changes to the starter code)
   5. The common LLVM tools are already installed and added to PATH inside the container, you can just call `clang`, `lli` or other tools from anywhere in your shell.

## Comments:

1. We highly recommend you use the environment provided to you. However, if you prefer to install the dependencies natively to your own environment. We use **LLVM ver. 15.0.7** and **gcc/g++ ver. 11.4.0**
2. The mounted directory containing the starter code for lab1 is visible both within the docker container and your host machine. You can do the editing of the src code from your laptop with your favorite code editor, and only build and run them within the docker container.