# Lab 0: Package Download and Environment Setup

## 1. Objective

The purpose of this lab is to check your environment settings and to make sure you can compile and run CUDA programs on the environment you’ll be using throughout the course. In this lab, you will:

* Get a copy of the assignment package and walk through the directory structure
* Set up the environment for executing the assignments
* Test the environment with a simple program that just queries which GPU device is attached

## 2. Preliminary work

**Step 1:** Clone the repository you will use for this assignment from the GitHub Classroom repository by using the link in the Canvas page for this assignment.

git clone <github-classroom-repository-url>

**Step 2**: Change into the new directory created by the cloning operation.

cd <directory-name>

You are now ready to begin the lab.

## 3. Configure the Makefile and Prepare to transfer it to the GPU server

**Step 1**: Modify the Makefile cloned from the repo at the two locations indicated by "TODO".

**Step 2**: Commit and push your changes to the Makefile to github. From the command line that will be something like *git add . ; git commit -m 'Changed makefile to be particular to this lab'; git push*

## 4. Transfer the code to the GPU server via github

**Step 1:** Use an **SSH program** to login to either artemis.emich.edu or apollo.emich.edu. Your home directory can be organized in any way that you like.

**Step 2:** Clone the github repository (now updated to include your changes to Makefile).

git clone < github-classroom-repository-url>

**Step 3**: Change into the new directory created by the cloning operation.

cd <directory-name>

## 5. Compile and run the lab

**Step 1:** Use the Makefile to build your executable by running the command

make

**Step 2**: Execute the lab by invoking the resulting executable. If your Makefile defined EXECUTABLE to be "device-query", you would use this command:

./device-query

You should expect see something like the following message:

There is 1 device supporting CUDA

Device 0: "Tesla T10 Processor"

Major revision number: 1

Minor revision number: 3

Total amount of global memory: 4294770688 bytes

Number of multiprocessors: 30

Number of cores: 240

Total amount of constant memory: 65536 bytes

Total amount of shared memory per block: 16384 bytes

Total number of registers available per block: 16384

Warp size: 32

Maximum number of threads per block: 512

Maximum sizes of each dimension of a block: 512 x 512 x 64

Maximum sizes of each dimension of a grid: 65535 x 65535 x 1

Maximum memory pitch: 2147483647 bytes

Texture alignment: 256 bytes

Clock rate: 1.30 GHz

Concurrent copy and execution: Yes

TEST PASSED