**Blue Waters Petascale Semester Curriculum v1.0**

**Unit 7: CUDA**

**Lesson 1: Introduction to CUDA GPGPU**

**Exercise Instructions for Students**

*Developed by Michael D. Shah for the Shodor Education Foundation, Inc.*



*Except where otherwise noted, this work by The Shodor Education Foundation, Inc. is licensed under CC BY-NC 4.0. To view a copy of this license, visit*[*https://creativecommons.org/licenses/by-nc/4.0*](https://creativecommons.org/licenses/by-nc/4.0)

*Browse and search the full curriculum at*[*http://shodor.org/petascale/materials/semester-curriculum*](http://shodor.org/petascale/materials/semester-curriculum)

*We welcome your improvements! You can submit your proposed changes to this material and the rest of the curriculum in our GitHub repository at*[*https://github.com/shodor-education/petascale-semester-curriculum*](https://github.com/shodor-education/petascale-semester-curriculum)

*We want to hear from you! Please let us know your experiences using this material by sending email to* [*petascale@shodor.org*](mailto:petascale@shodor.org)

Welcome students to an introduction to CUDA!

* We are going to talk a bit about the GPU architecture
* I am going to show you how to execute your first GPU program.

Installing CUDA

1. Students should navigate to the NVidia website to download the latest toolkit.

* At the time of writing it is version 11.0 and located here: <https://developer.nvidia.com/cuda-downloads>
* For linux it may be as simple as typing: `sudo apt-get install nvidia-cuda-toolkit`

1. If students are working on their own machines, they may also want to verify they have a CUDA enabled card/

* They may [google this information](https://www.google.com/search?q=is+my+card+cuda+enabled&oq=is+my+card+cuda+ena&aqs=chrome.0.0j69i57j0.3126j1j7&sourceid=chrome&ie=UTF-8) OR
* Check here: <https://developer.nvidia.com/cuda-gpus>

Student Exercise

Some sample code has been provided along the lecture. Students, now it is your chance to run that code.

1. Compile and run each of the hello\_cpu.c and hello\_gpu.cu codes.
2. Time the difference between the CPU and GPU code using time\_t structures. Alternatively you can use `time` Unix command.
3. Why might the CPU code run faster than the GPU code in this example?

* Think about and write down what we must spend time doing and where any 'latency' is in terms of time when doing CUDA programming.
* Think about and write down if GPU programs will **always** run faster than CPU programs.