## WebGPU Tutorial

We will be using a web based lab development and submission system through this course. The purpose of this web system is two fold:

- 1. it allows you to develop CUDA applications without the need to buy and/or setup a CUDA system, and
- 2. it allows us to grade your submissions more efficiently, uniformly, and promptly.

This document describes the online system, walking through the pages that you will see, and describes a sample submission. The submission system is designed to support the entire development and submission cycle for all the lab assignments. While you have to use the submission system to receive grades for this course, we will allow you to develop locally and use the submission system after you are reasonably satisfied with your code.

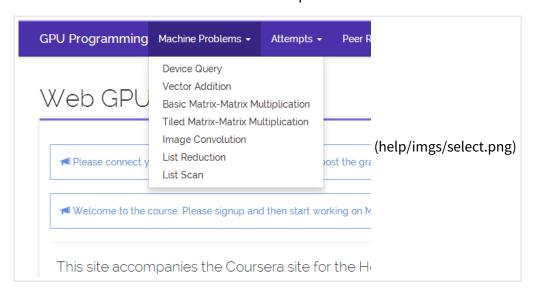
# Web System

This section describes the interface and system you will be using for this course.

## Development Interface

Once you have signed up and logged into the website, you can start your development. You can access the machine problems (MP) by using the **Machine Problem** tab.

Here we select the first MP from the **Machine Problems** dropdown.



webgpu.com/help 1/10

You will then be able to see the description of the MP. This describes objective, instructions, deadlines, and the grading policy.



The MP page shows the description, code, questions to be answered, previous attempts, and program history. If you click the **Code** tab, you can start developing the MP.

```
CODE

Code

Compile & Run

Code

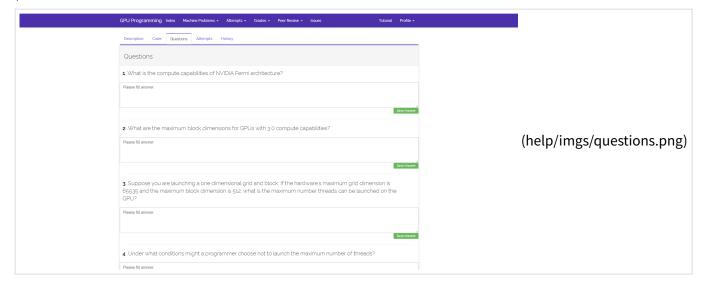
Compile & Run

Com
```

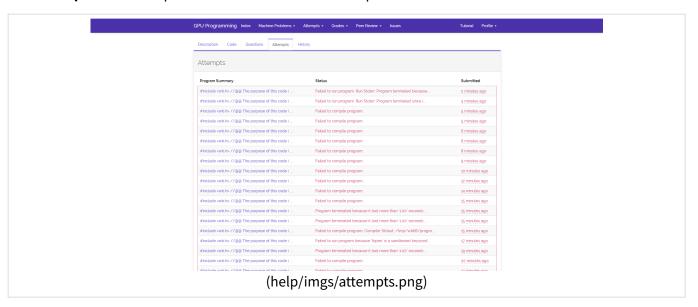
Once you have a solution, you click on the **Compile & Run** tab. This will first save your work on the server and then submit it for processing (note: while code is saved automatically by the system, it is a good idea to keep a backup of the program). In case the server is overloaded, your computation will be placed in a queue (this is a great motivation to not start the lab at the last minute).

The **Questions** tab shows questions to be answered for this MP.

webgpu.com/help 2/10



#### The **Attempts** tab shows previous code submission attempts



#### **Insertion Points**

Most of instructions for the lab are found in the code. We use //@@ to demarcate where code needs to be inserted. In lab 1, for example, you see code such as:

```
wbTime_start(GPU, "Copying input memory to the GPU.");
//@@ Copy memory to the GPU here
wbTime_stop(GPU, "Copying input memory to the GPU.");
```

This means that you need to insert a CUDA memory copy operation where the comment is located.

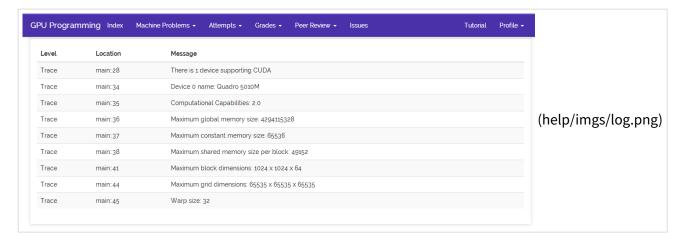
## Logging and Debugging

Logging is facilitated through the use of a logging API. The logging function wbLog takes a level which is either OFF, FATAL, ERROR, WARN, INFO, DEBUG, or TRACE and a message to be printed. To print the value of variable x, for example you type wbLog(TRACE, "The value of x = x). Note that

webgpu.com/help 3/10

variables are serialized to a string when placed in the wbLog function, so you can place reals, integers, and strings in the logging function.

The result of the logging messages is shown in the Attempts tab once you submit your code.



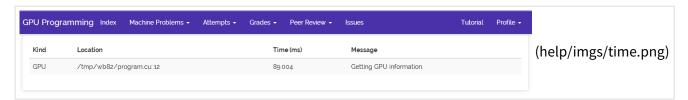
Debugging is done in the form of logging, so you cannot place break points, etc...

## Timing Code

Logging is facilitated through the use of a timing API. The timing functions wbTime\_start and wbTime\_stop . To time a section of code, you write wbTime\_start(tag, msg) and then wbTime\_stop(tag, msg) , where the tag and msg must match. In the template code given for Lab 1, for example, we do

```
wbTime_start(GPU, "Allocating GPU memory.");
//@@ Allocate GPU memory here...
wbTime_stop(GPU, "Allocating GPU memory.");
```

the above would measure the time it takes to allocate GPU memory. The timing result is shown in the **Attempts** tab after you submit the program.



The timing information includes the tag and message you specified when you timed your code, and also which line you are timing. Since it is important to understand the performance of your code, it is valuable to spend time to understand the output of the timer.

#### Previous Attempts

All previous attempts are saved and recorded on the server side. To be graded for an attempt, you must click the grade button when that attempt is displayed.

# Behaviors

webgpu.com/help 4/10

While developing the labs, you may encounter one or more of the following behaviors.

#### No Error

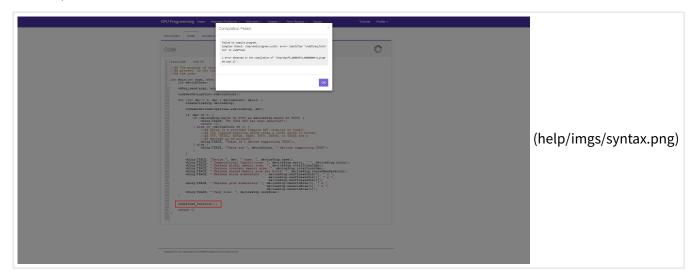


This means that no errors were found while compiling and running the program. Your program has been checked against the expected solution of the dataset you selected. Note that it might be the case that the program may run correctly on one dataset and not the other. You should check that your program runs on all datasets before submitting a program for grading.

#### Solution is Incorrect

This means that while the program did compile and run without errors, the solution did not match the expected results. This problem could stem from either having an implementation error in your algorithm, or from having incorrect logic in the host code. Logging the state of your program at different points would help you debug the problem.

## Compilation Failed



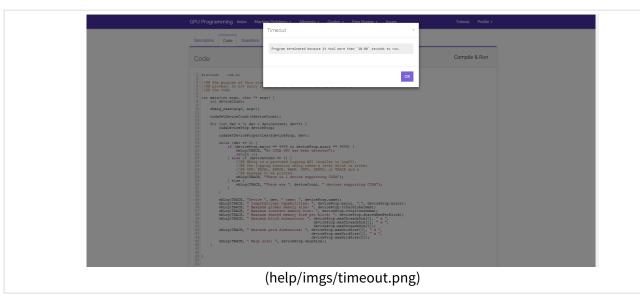
webgpu.com/help 5/10

This error message occurs when the program submitted failed to compile. The output from the compiler is shown in the error window. The line number of where the compilation error occurs is given in the error message.

### **Attempt Limiter**

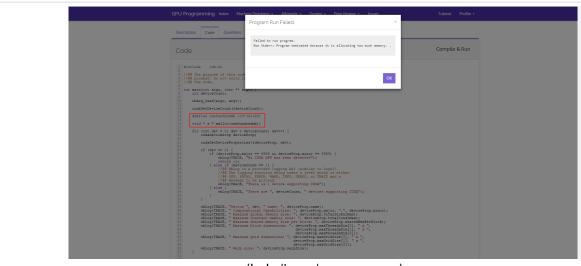
This error occurs when you submit more than one program within 10 seconds of each other. To maintain stability, users can only submit one program every 10 seconds.

## Program Terminated due to Timeout



The most likely cause of this error is that you have inadvertently placed an infinite loop either in your CPU or GPU code. Part of the reason for this behavior is that the system ensures fairness (i.e. you should not hog the machine). To ensure fairness, the system is configured to terminate long running processes. When you see this error, you have hit that timeout limit.

## Memory Allocation Error

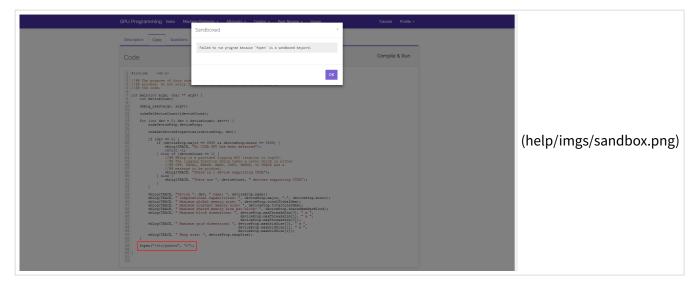


(help/imgs/memory.png)

webgpu.com/help 6/10

To maintain system stability, users are not allowed to allocate too much memory.

### Sandboxing Error



To maintain security, we run your program in a sandboxed mode. This means that you are restricted to using our API functions for certain tasks, rather than using system calls or C library functions. The sandbox error will sometimes tell you what call is being caught, although that might not always be the case.

## Machine Problem Questions

The machine problem questions are meant to allow you to think and explore your code further. Unlike the code section, it is enough to save your answer to submit the questions for reviewing.

Answers to the questions are not graded and we encourage you to discuss them on the forums and through your study groups.

# **Development Suggestions**

The following suggestions are recommended when getting started using the system

- Develop your application incrementally
- Do not modify the template code provided only insert code where the //@@ demarcation is placed
- Make sure to test your results against all the datasets provided
- Do not wait until the last minute to attempt the lab.

### Further Tips

- Code is autosaved every 3 seconds and you can see the history in the history tag. Again, do keep a
  backup of your programs. We suggest you backup using a private source code repository. Free git
  private repositories are offered by Github and Bitbucket.
- You can reset the MP (get the original skeleton code) by going to webgpu.com/mp/1/revert (replacing the 1 with the mp number) and then back to the MP page

webgpu.com/help 7/10

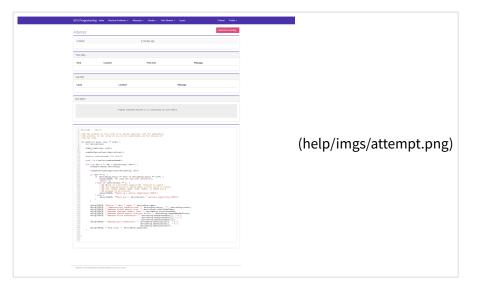
• Tab completion is supported in the code editor by using the Ctrl+Space key combination

Ctrl+S saves your code along with the answers to the questions to the server

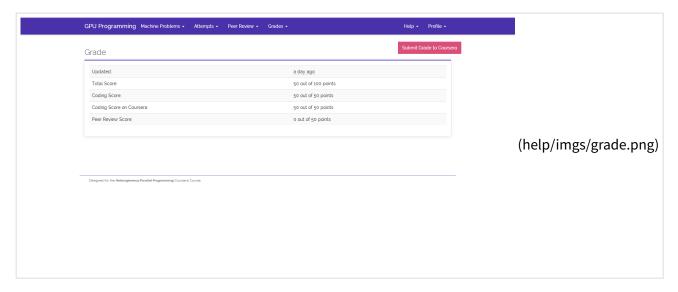
## Grading

Grading is performed based on criteria that are specific for each lab.

For each attempt in the attempts tab, you'll see a submit for grading button.



Clicking on that would tell the system to grade that attempt and redirect you to the grade report page.



The grades are automatically sent to Coursera. You will have to submit your grade back to Coursera by clicking the button.

# Local Development

While not required, the library used throughout the course can be downloaded from Github (https://github.com/abduld/libwb). The library does not depend on any external library (and should be cross platform), you can use make to generate the shared object file (further instructions are found on the Github

webgpu.com/help 8/10

page). Linking against the library would allow you to get similar behavior to the web interface (minus the imposed limitations). Once linked against the library, you can launch the your program as follows:

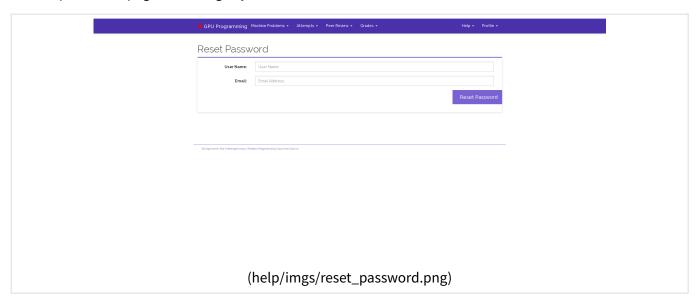
```
./program -e <expected_output_file> -i <input_file_1>,<input_file_2> -o <outpu
t_file> -t <type>
```

The <expected\_output\_file> and <input\_file\_n> are the input and output files provided in the dataset. The <output\_file> is the location you'd like to place the output from your program. The <type> is the output file type: vector, matrix, or image. If an MP does not expect an input or output, then pass none as the parameter.

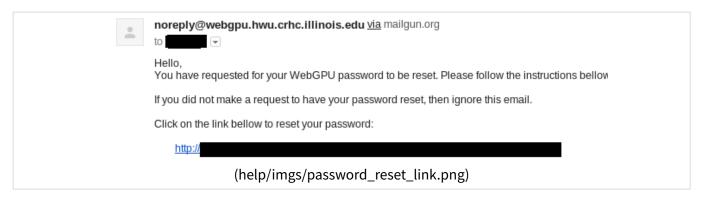
In case of issues or suggestions with the library, please report them through the issue tracker (https://github.com/abduld/libwb/issues) on Github.

## Resetting Password

If you have forgotten your account password, then you can reset it by going to the password reset (/reset\_password) page and filling in your user name and email.

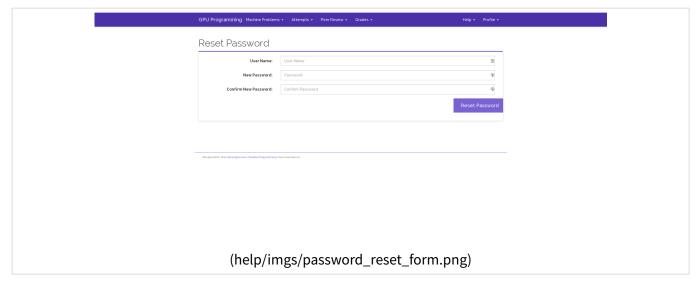


You will then be emailed a link that you can use to reset your password.



Which will direct you to the password reset form

webgpu.com/help 9/10



# Plagiarism

If we detect any plagiarism in the code or answers, the two parties will get 0 as their grade for the MP for the first offense. Multiple offenses will result in a 0 for the entire course.

# System Requirements

A recent web browser is the only requirement for using and submitting labs in this course. We have mainly tested the website using the Google Chrome browser.

# Issues/Questions/Suggestions

In case of questions, please post them to the forums. For issues or suggestions, please use the issue tracker (https://github.com/abduld/wb/issues).

Designed and architected by Abdul Dakkak (https://www.dakkak.dev/).

webgpu.com/help