Page 7, 9. What is a compiler? NVCC and MS Visual C / gcc

Page 11. What is a pointer:

A variable that holds the address of another variable.

Referencing a pointer (put the address of a var in a pointer):

int c;

int\* p;

c = 5;

p = &c;

Dereferencing a pointer (get the value of the var whose address the pointer holds):

int n;

n = (\*p);

In C/C++ a pointer also stands for a dynamically allocated array. Do int\* p\_array, than allocate some memory then do p\_array[i].

Page 13. That is what malloc(), free() and memcopy() do.

Pages 14-16. Make a new CUDA project. Compile the example. Add to the main() a printf statement to see the results

Page 15. What is this horrible (void\*\*)&dev\_a parameter type:

The CUDA API passes all values by reference. C does not allow this. So we need to pass by value the address of the variable (&dev\_a) that we want to pass as a parameter explicitly (which is what passing by reference actually does under the hood). But the variable we want to pass (dev\_a) is itself a pointer (i.e. holds the address of another variable). So the type of the thing that we pass as a parameter needs to be an address of an address, i.e. a double pointer (void \*\*)&dev\_a. The void here means that we don’t care about the actual type of the original variable. Why is it ok not to care?

Pages 21, 22. Compile this example. Use random\_ints from the repo. Put everything that is in the main() in another function (e.g. calling\_add\_kernel()) and make a main() that just calls that.

Pages 25-27. Compile this (follow the yellow letters, i.e. threadId.x and <<<1,N>>>)

Pages 31-33. As above.

Pages 45-47. As above. Add another helper function (calling\_dot\_kernel()) and call this from the main.

Page 53. The jumble at the bottom should read \*c += sum;

Pages 58-60. Compile this last example

And now for the fun bit. If there is time left go to

http://numba.pydata.org/numba-doc/dev/cuda/index.html

<https://people.duke.edu/~ccc14/sta-663/CUDAPython.html>

<https://docs.continuum.io/numbapro/CUDAJit> (older but good overall example)

Read the numba, CUDA tutorials and try to write the last example in python.