

# Hands On OpenCL

## An open source two-day lecture course for teaching and learning OpenCL

### Welcome

Hands On OpenCL is a two-day lecture course introducing OpenCL, the API for writing heterogeneous applications. Provided are slides for around twelve lectures, plus some appendices, complete with Examples and Solutions in C, C++ and Python. The lecture series finishes with information on porting CUDA applications to OpenCL.

This set of freely available OpenCL [exercises and solutions](#), together with [slides](#) have been created by Simon McIntosh-Smith and Tom Deakin from the University of Bristol in the UK, with financial support from the Khronos Initiative for Training and Education ([KITE](#)) to promote the use of open standards.

[Simon McIntosh-Smith](#) is one of the foremost OpenCL trainers in the world, having taught the subject since 2009. He has run many OpenCL training courses at conferences such as SuperComputing and HiPEAC, and has provided OpenCL training for the UK's national supercomputing service and for the Barcelona Supercomputing Center. With OpenCL training experience ranging from half day on-site introductions within companies, to two-day intensive hands-on workshops for undergraduates, Simon can provide customized OpenCL training to meet your needs. Get in touch if you'd like to know more:

For more about the authors, please visit [Simon's home page](#) or [Tom's home page](#).

These lectures, and their examples, and released under the "attribution CC BY" creative commons license. In other words, you can use these in any way you see fit, including commercially, but please retain an attribution for the original authors, Simon McIntosh-Smith and Tom Deakin.

### Get the slides and code

The slides are available under [Releases](#). The code is available in the [Exercises and Solutions repository](#).

### Course Structure

1. Introduction to Heterogeneous Parallel Computing

- Setting up your OpenCL environment (AMD, Intel, NVIDIA)

2. An overview of OpenCL

3. Important OpenCL concepts

- Platforms, contexts, programs, queues, buffers and kernels

- NDRanges, Work-Groups, Work-Items

4. Overview of OpenCL APIs

- C, C++ and Python

5. Introducing OpenCL kernel programming

6. Understanding the OpenCL memory hierarchy

7. Synchronization in OpenCL

- Events and barriers

8. Heterogeneous computing with OpenCL

- Using CPUs and GPUs simultaneously, multiple platforms and devices

9. Enabling portable performance via OpenCL

- Autotuning using Flamingo

10. Optimizing OpenCL performance

- Profiling using Extrae and Paraver Information on NVVP and CodeXL

11. Debugging OpenCL

## Using GDB

## 12. Porting CUDA to OpenCL

## Examples

Download the examples by checking out the git repository with the command:

```
git clone git://github.com/HandsOnOpenCL/Exercises-Solutions.git
```

### 1. Platform Information

Run a simple OpenCL program to give you some key facts about the devices available in your system.

### 2. VADD - The OpenCL "Hello World"

Start by looking at the C API for this program which introduces the OpenCL computational model.

### 3. VADD - Now in C++ and Python

### 4. Chaining vector add kernels

Extend VADD to compute  $C=A+B$ ;  $D=C+E$ ;  $F=D+G$  by running the kernel multiple times.

### 5. Extend VADD for $D = A + B + C$

Extend the VADD kernel to compute a different sum.

### 6. Matrix Multiplication

Write your first OpenCL kernel from scratch.

### 7. Using private memory

Use private memory to minimize memory costs.

### 8. Using local memory

Use local and private memory to minimize memory costs.

### 9. The Pi program

Estimate Pi by integration.

### 10. Heterogeneous Computing

Run your kernels on many devices.

### 11. Optimize matrix multiplication

Look at portable performance (combining 9. and 10.)

### 12. Profiling OpenCL programs

Experiment making things run faster.

### 13. Porting CUDA to OpenCL

Convert a simple CUDA application to OpenCL (program TBA).

## Authors and Contributors

Simon McIntosh-Smith, University of Bristol

Tom Deakin ([@tomdeakin](#))

## Support or Contact

Found a bug or wish to suggest an update to the material? Please submit a new Issue in the relevant repository ([Exercises](#) or [Slides](#))

Fixed a bug yourself? Please submit a [pull request](#). Thanks.

This page was generated by [GitHub Pages](#) using the [Cayman theme](#) by [Jason Long](#).