# FDD3360 Applied GPU Programming 7.5 credits

## Tillämpad GPU-programmering

This is a translation of the Swedish, legally binding, course syllabus.

If the course is discontinued, students may request to be examined during the following two academic years

## Establishment

Course syllabus for FDD3360 valid from Spring 2019

## Grading scale

P, F

## Education cycle

Third cycle

## Specific prerequisites

PhD students at KTH

## Language of instruction

The language of instruction is specified in the course offering information in the course catalogue.

## Intended learning outcomes

After passing the course, the student will be able to:

- analyze the GPU architecture, assess its advantages and identify potential software optimizations based on the knowledge of the GPU architecture.
- design and implement a computer code for GPU with application to scientific computing, machine learning, image and video processing, computer graphics or mobile programming.
- experimental high-productivity approaches for GPU programming, such as GPU libraries and computing frameworks, to speed-up the development of large GPU applications.
- use effectively development tools for GPU programming, such as debuggers and performance monitoring tools.
- prepare a written report on the design, development and implementation of a code for GPU (with application to scientific computing, machine learning, image and video processing, computer graphics or mobile programming) and present orally the report during a seminar.

## Course contents

The course focuses on three main topics:

- GPU architecture. The computing and memory systems of different commercial GPUs are introduced. A comparison with conventional CPU and presentation of new upcoming GPUs will be given.
- GPU programming with CUDA. The CUDA concepts and how to use them to develop applications for GPU are introduced by making examples from different fields, such as image processing or scientific computing. Also development tools, such debuggers and performance monitoring tools are presented.
- GPU programming with GPU libraries and frameworks. High-productivity computing frameworks, among which the Thrust library, OpenACC and cuDNN, are presented. Different frameworks will be explained by providing examples from different computer science areas.

Students will be given access to the GPU cluster, Tegner, at PDC if they do not have access to a computer with GPU.

## Examination

- EXA1 - Examination, 7.5 credits, grading scale: P, F

Based on recommendation from KTH's coordinator for disabilities, the examiner will decide how to adapt an examination for students with documented disability.

The examiner may apply another examination format when re-examining individual students.

The examination consists of:

- LAB1 - Laboratory assignment, 1 , grade: P, F
- LAB2 - Laboratory assignment,, 1, grade: P, F
- LAB3 - Laboratory assignment, 1, grade: P, F

- PRO1 - Project work, 4,5, grade: P, F

The project report must include a "related work" section, presenting a literature survey of the topic.

## Other requirements for final grade

Passing grade on all sections: LAB1, LAB2, LAB3, PRO1

## Ethical approach

- All members of a group are responsible for the group's work.
- In any assessment, every student shall honestly disclose any help received and sources used.
- In an oral assessment, every student shall be able to present and answer questions about the entire assignment and solution.