# SCIT

**School of Computing & Information Technology**

## CSCI376 – Multicore and GPU Programming
## Spring 2022

## Assignment 1

**Due on Friday, 26th August 2022 at 17:00 AEST**

**Task**

Write a program using OpenCL that does the following:

- Search the system for all CPU and GPU devices (note that some systems have multiple CPUs and GPUs). Allow the user to enter whether he/she wants to use a CPU or a GPU device.

  (1 mark)

- Based on the user's choice, display the following information for each CPU/GPU device available on the system:
  - Name of the platform(s) that support that device
  - Device type – CPU or GPU

    (hint: CL_DEVICE_TYPE_CPU or CL_DEVICE_TYPE_GPU)
  - Device name
  - Number of compute units
    (hint: CL_DEVICE_MAX_COMPUTE_UNITS)
  - Maximum work group size
    (hint: CL_DEVICE_MAX_WORK_GROUP_SIZE)
  - Maximum work item sizes
    (hint: CL_DEVICE_MAX_WORK_ITEM_SIZES)
  - Global memory size
    (hint: CL_DEVICE_GLOBAL_MEM_SIZE)

  (1 mark)

- Based on the devices available, allow the user to select one device. Check whether that device supports the cl_khr_fp16 extension (hint: CL_DEVICE_EXTENSIONS). Create a context and a command queue for that device.

  (1 mark)

- Read the program source code from the provided “source.cl” file and build the program. Display whether or not the program was built successfully and display the program build log (display the build log even if the program was built successfully. Note: for some compilers, the build log will be empty if successful).

  (1 mark)

- Find and display the number of kernels in the program. Create kernels from the program and display all the kernel function names.

  (1 mark)

## Screenshots

Include screenshots with your submission to demonstrate that the program works on your computer. The screenshots should capture the input and output of your program. Use one of the common image formats (i.e. jpg/png/bmp).

## Instructions and Assessment

Zip all your **source files** (.cpp, .h and .cl files) and the **screenshots** (.bmp/.jpg/.png) into a single file and submit this via Moodle by the due date and time (do **NOT** zip your entire project file as this can be very large, and do **NOT** use .rar). Assignments that are not submitted on Moodle will not be marked.

The assessment must be your own work. If asked, you must be able to explain what you did and how you did it. Marks will be deducted if you cannot correctly explain your code.

NOTE: The mark allocations shown above are merely a guide. Marks will be awarded based on the overall quality of your work. Marks may be deducted for other reasons, e.g., if your code is too messy or inefficient, if you cannot correctly explain your code, etc.

For code that does not compile, does not work or for programs that crash, the most you can get is half the marks (i.e., 2.5 marks or less). It is better to comment out sections of your code that do not work and include a note for the marker.