THE UNIVERSITY OF WINNIPEG

Applied Computer Science

Course Number: **GACS-7306-001**
Course Name: **Applied Parallel Programming**
Course Webpage: http://courses.acs.uwinnipeg.ca/4306-001/

Instructor Information

Instructor: Dr. Christopher Henry
Email: ch.henry@uwinnipeg.ca

Class Room No: 3D03
Class Meeting Time: M/W 2:30 - 3:45 pm
Office Hours: Wednesday 1:30-2:30 pm

Important Dates

First Class:	September 6th, 2017
Fall Reading Week:	October 8th – 14th, 2017 (No classes)
Project Proposal Presentation:	October 16th, 2017
Midterm Test:	October 25th, 2017
Withdrawal date w/o academic penalty[2]:	November 10th, 2017
Last Scheduled Class:	December 4th, 2017
Final Project due date and Presentation:	December 11th, 2017
Final Test (Comprehensive):	December 18th, 2017 1:30 – 4:30 pm
The University is closed on the following dates:	
	October 9th, 2017
	November 11th, 2017
	December 22nd, 2018 – January 2nd, 2018

Note:
The lecture on October 25th may be rescheduled to December 5th or 6th. Details will be discussed in class.

[2]A minimum of 20% of the work on which the final grade is based will be evaluated and available to the student before the voluntary withdrawal date.

Course Objectives/Learning Outcomes

The basis of this course is the parallel execution model, which is a generalization of the traditional single threaded paradigm. The focus is parallel and distributed computing for use in high-performance scientific applications. Students gain considerable knowledge in multi-core processors, concurrency, parallel execution, latency, communication and coordination among processes, message passing, shared-memory models, optimization techniques, parallel algorithms, decomposition strategies, system architecture, and performance analysis and tuning. Using the language C/C++, students gain hands-on experience writing scalable parallel applications for Graphics Processing Units.

Evaluation Criteria

Midterm Examination (30%)

There will be **one** midterm test.

Project Proposal (5%)

The aim of this project is to solve an interesting scientific problem by creating a high-performance parallel application. The best way to identify a project topic is to choose a domain of interest (preferably in your research area). Create a proposal explaining the problem domain you have selected. Include a description of the theoretical framework you will be implementing. Also, indicate the topics or papers you are considering for your implementation and the dataset you will be working with. Please schedule a consultation with the instructor to review your selection and determine its suitability. Submit a brief project proposal (max. 5 pages) The project proposal may contain the following sections (not necessarily all).

- Title page (including project title)
- Introduction and project idea
- Background information
  - Literary search
- Problem description
- Theoretical framework
  - Topics and/or papers
- Requirement definitions and functional specifications
- Proposed System
- Proposed System Architecture
  - Software Architecture
  - Hardware Architecture
- Development Environments
- Proposed test and/or experiments
  - Data set
- Proposed verification and validation plans

Project Proposal Presentation (5%)

Project presentation should cover the salient points from your report and is generally no longer than 10 minutes in length.

Final Project Report (20%)

The final report may contain the following sections.

- Title page (including project title)
- Abstract
  - 200 words or less
  - Keywords
- Introduction
  - Motivation, problem definition, contribution/outcome
- Background
  - Relevant information to understand your problem, related works

- Theoretical Framework
    - Detailed description of the theory used in your project
    - Description should start with fundamental concepts/defns and build upon this foundation to the framework used in your project
- Implementation
    - System description
        - Software architecture
        - Hardware architecture
    - Development environments
    - Code and design diagrams
    - Algorithm description
        - Complexity analysis
- Experiments and Results
    - Details of the experiments and your observations
- Analysis
- Conclusion
- References

Students must include all source code and executables with their final report. The will code will not be marked, but will be used to verify the results in the report.

Final Project Presentation (20%)

The proposal presentation should cover the salient points from your final report. The presentation is ~20 minutes in length, followed by a 5 minute question and answer period.

Final Quiz (20%)

The final quiz is comprehensive.

Final Letter Grade Assignment

Historically, numerical percentages have been converted to letter grades using the following scale. However, instructors can deviate from these values based on pedagogical nuances of a particular class, and final grades are subject to approval by the Department Review Committee.

A+	90+ - 100%	B	70 - 74%	F	below 50%
A	85 - 90%	C+	65 - 69%		
A-	80 - 84%	C	60 - 64%		
B+	75 - 79%	D	50 - 59%		

Exam Requirements

- Photo ID is required
- Unless a medical certificate is provided, no accommodation is made for missed exams
- No equipment (*e.g.* calculators, dictionaries, handheld devices) are authorized for use in tests/exams

Student Services and Information

Students with documented disabilities, temporary or chronic medical conditions, requiring academic accommodations for tests/exams (*e.g.*, private space) or during lectures/laboratories (*e.g.*, note-takers) are encouraged to contact Accessibility Services (AS) at 786-9771 or accessibilityservices@uwinnipeg.ca to discuss appropriate options. All information about a student's disability or medical condition remains confidential http://www.uwinnipeg.ca/accessibility.

All students, faculty and staff have the right to participate, learn, and work in an environment that is free of harassment and discrimination. The UW Respectful Working and Learning Environment Policy may be found online at www.uwinnipeg.ca/respect

Students may choose not to attend classes or write examinations on holy days of their religion, but they must notify their instructors at least two weeks in advance. Instructors will then provide opportunity for students to make up work examinations without penalty. A list of religious holidays can be found in the 2017-18 Undergraduate Academic Calendar.

Required Textbooks

Main texts:

- D. B. Kirk, and W. W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*. USA: Elsevier, 2013
- H. Nguyen, Ed., *GPU Gems 3*. USA: Addison-Wesley, 2008. [Online]. Available: NVIDIA, https://developer.nvidia.com/content/gpu-gems-3.

Besides the information contained in the main texts, I may also distribute papers, and discuss appropriate material and examples from other sources. Students are responsible for all material covered in the class.

Prerequisite Information (This information can be found in the UW General Calendar)

- Consent of Department Graduate Studies Committee Chair
- Students who have taken ACS-4306 will not be eligible to take ACS-7306 for degree credit.

Misuse of Computer Facilities, Plagiarism, and Cheating

Academic dishonesty is a very serious offense and will be dealt with in accordance with the University's discipline bylaw. Be sure that you have read and understood **Regulations & Policies #8**, starting on page 9, in the 2017-2018 UW Course Calendar available at: http://www.uwinnipeg.ca/index/calendar-calendar.

Students are strongly recommended to view the University of Winnipeg library video tutorial *Avoiding Plagiarism*, which is available at: https://www.youtube.com/watch?v=UvFdxRU9a8g

Course Topics

1. Course Introduction
2. History of GPU Computing

3. Introduction to Data Parallelism and CUDA C
4. Data-Parallel Execution Model
5. CUDA Memories
6. Performance Considerations
7. Floating-Point Considerations
8. Parallel Patterns:
    a. Convolution
    b. Prefix Sum
    c. Sparse Matrix-Vector Multiplication
9. Case Studies:
    a. Advanced MRI
    b. Molecular Visualization and Analysis
10. Parallel Programming and Computational Thinking
11. Heterogeneous Computing Clusters
12. Dynamic Parallelism

Note: not all of the above topics may be covered.

Course Readings

Relevant textbook chapters and sections will be given during lectures.

Recommended Study Habits

Students who do well in this class tend to spend an extra 3-5 hours per week doing the following:

- Read the textbook before coming to class
- Attend lectures
- Take notes
- Attempt the problems and exercises at the end of the chapters
- Submit all the exercises and assignments
- Form study groups to study for the midterm and exam
- Regularly ask questions

Advice: Students who fall behind find it very hard to catch up.