Bilkent University

Department of Computer Technology and Information Systems

2016 - 2017 SPRING Semester

Course Code	**CTIS 484**
Course Name	**Advanced Topics in Programming**
Course Credit	**4 (4 hours Lecture + Lab)**
Instructor	**Okyay SAY** **office: C213** **e-mail:** okyaysay@bilkent.edu.tr
Description	Object oriented programming concepts using C++ programming language. Understanding Parallelism with GPUs. CUDA Architecture. Developing applications for processors with parallel computing resources. Fundamental concepts and in-depth knowledge about parallel, distributed, grid and cloud computing programming principles, communication models, memory utilization and limitations of these processors. Atomic operations. Using single and multiple Streams. Performance measuring.
Text Book	C++ How to Program, Deitel & Deitel, Pearson CUDA By Example, Jason Sanders & Edward Kandrot, Addison-Wesley
Other Materials	Lecture notes by Okyay Say: "Object Oriented Programming using C++" Lecture notes by Okyay Say: "Parallel Programming using CUDA C++" CUDA Programming A Developers Guide to Parallel Computing with GPUs, Shane Cook, Morgan Kaufmann CUDA Application Design and Development, Rob Farber, Morgan Kaufmann Programming Massively Parallel Processors, D. B. Kirk, W. W. Hwu, Morgan Kaufmann
Important Notes	Students who do not attend at least 50% of the lectures will receive an "FZ" letter grade. If a student's overall grade (calculated by saps from assignments, homeworks, projects, quizzes and midterm) is less than 25.0 (25/70) on the last day of classes, (s)he will receive an "FZ" letter grade and will not be able to attend to final or retake exam.

Assessment

Assignments	**20%**
Homeworks	**6%**
Projects	**14%**
Quizzes	**10%**
Midterm	**20%**
Final	**30%**

Grading Criteria

A	**[86 – 100]**
A-	**[80 – 86)**
B+	**[75 – 80)**
B	**[70 – 75)**
B-	**[65 – 70)**
C+	**[61 – 65)**
C	**[57 – 61)**
C-	**[53 – 57)**
D+	**[49 – 53)**
D	**[45 – 49)**
F	**[0 – 45)**

IMPORTANT NOTE The Information contained in the course syllabus is subject to change. Students will be informed of any changes either in the class and/or by e-mail and/or publishing at the course moodle page. It is each student's responsibility to regularly check the course's moodle page and his/her Bilkent e-mail account to learn about any course related changes and announcements.

DETAILED COURSE OUTLINE

Week	Lecture Topics	Lab. Objectives
1 **6 Feb. – 10 Feb.**	General Information about the course Introduction to C++ Scope resolution operator Reference variables, const member functions Static members Constructor initializer, Default function arguments	**NO LAB**
2 **13 Feb. – 17 Feb.**	Dynamic memory, Destructor, Member-wise copy, Copy constructor	**Lab 1:** Classes containing dynamic memory operations
3 **20 Feb. – 24 Feb.**	Friend functions, friend classes Operator overloading, non-member operator overloads Friend operator overloads, member operator overloads The this operator, Prefix and postfix notations	**Lab 2:** Operator overloading
4 **27 Feb. – 3 Mar.**	Inheritance, Single inheritance, Layered classes Derived class constructors, destructors, Visibilities Multiple Inheritance, Ambiguity resolution	**Lab 3:** Inheritance
5 **6 Mar. – 10 Mar.**	Virtual base classes Inheritance hierarchy and pointers Runtime polymorphism, Abstract base classes.	**Lab 4:** Polymorphism
6 **13 Mar. – 17 Mar.**	Templates I/O header files, file operations I/O of objects	**Quiz-1**
7 **20 Mar. – 24 Mar.**	Introduction to parallel and distributed computing. Introduction to basic CUDA concepts and the tools needed to build and debug CUDA applications.	**Lab 5:** Distinguishing CUDA from Conventional Programming with a simple example.
8 **27 Mar. – 31 Mar.**	Difference between host code and device code.	**Lab 6:** Writing first CUDA C++ program.
9 **3 Apr. – 7 Apr.**	Device memory usage. Querying system information on CUDA-capable devices. Fundamental ways CUDA exposes parallelism. Two-dimensional grids.	**MIDTERM EXAM**
10 **10 Apr. – 14 Apr.**	Thread cooperation. How to synchronize the parallel execution of different threads.	**Lab 7:** First parallel code with CUDA.

11 **17 Apr. – 21 Apr.**	Performance measuring.	**Lab 8:** Thread cooperation.
12 **24 Apr. – 28 Apr.**	How to use constant memory.	**Quiz-2** **Lab 9:** Matrix operations in parallel.
13 **1 May – 5 May**	How to use texture memory.	**Lab 10:** Performance differences when different number of blocks and threads are used.
14 **8 May – 12 May**	Atomics, Streams.	**Lab 11:** Constant memory examples