

# 확장강의계획서

## (2025년도 1학기)

과목명	기초 GPU 프로그래밍	과목번호	CSEG483/CSE5483
구분(학점)	3	수강대상	“2. 선수학습내용” 조건을 만족하는 학부생
수업시간	화,목 16:30~17:45	강의실	

담당교수	성명: 임 인 성	홈페이지: grmanet.sogang.ac.kr/~ihm
	E-mail: ihm@sogang.ac.kr	연락처: 02-705-8493
	장소: AS905 면담시간: 추후 공고	

### I. 교과목 개요(Course Overview)

1. 수업개요																
<p>GPU(Graphics Processing Unit)는 CPU와 함께 컴퓨팅 시스템을 구성하는 핵심 프로세서 중의 하나이다. 실시간 3D 컴퓨터 그래픽스 계산의 가속을 위하여 놀라운 속도로 발전해온 GPU는 현재 슈퍼 컴퓨팅, 인공지능, 그리고 빅 데이터 등의 대용량 데이터의 고속 병렬 처리가 필요한 여러 분야에서 널리 사용되고 있다. 따라서, GPU를 최적으로 활용할 수 있는 GPU 프로그래밍능력은 C/C++ 프로그래밍과 같은 CPU 상에서의 전통적인 프로그래밍능력과 함께 고급 소프트웨어 개발자가 갖춰야 할 기본 소양으로 자리 잡아가고 있다. 본 과목에서는 대표적인 GPU 프로그래밍 툴인 CUDA(Compute Unified Device Architecture)를 기반으로 하여, SIMT(Single Instruction, Multiple Threads) 구조의 GPU가 제공하는 강력한 병렬 처리 성능을 효과적으로 활용할 수 있는 GPU 프로그래밍 기술에 대하여 학습한다.</p>																
2. 선수학습내용																
<ul style="list-style-type: none"> <li>- <b>프로그래밍:</b> 컴퓨터공학과 3학년 1학기생 수준의 C/C++ 프로그래밍 능력</li> <li>- <b>이론:</b> 자료구조/알고리즘/컴퓨터 시스템 등 컴퓨터공학과 3학년 1학기 수준의 컴퓨터공학 지식</li> </ul> <p><b>[참고]</b> 본 과목은 <b>학석사연계과목</b>이며, 강의 수준은 위에서 기술한 <b>선수학습능력을 갖춘 컴퓨터공학과 3학년 생 이상의 수강생들이 따라올 수 있는 수준으로 진행할 예정임.</b></p>																
3. 수업방법 (%)																
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 16.6%;">강의</th> <th style="width: 16.6%;">토의/토론</th> <th style="width: 16.6%;">실험/실습</th> <th style="width: 16.6%;">현장학습</th> <th style="width: 16.6%;">개별/팀 별 발표</th> <th style="width: 16.6%;">기타</th> </tr> </thead> <tbody> <tr> <td>100%</td> <td>%</td> <td>0%</td> <td>%</td> <td>%</td> <td>%</td> </tr> </tbody> </table>	강의	토의/토론	실험/실습	현장학습	개별/팀 별 발표	기타	100%	%	0%	%	%	%				
강의	토의/토론	실험/실습	현장학습	개별/팀 별 발표	기타											
100%	%	0%	%	%	%											
4. 평가방법 (%)																
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 12.5%;">중간고사</th> <th style="width: 12.5%;">기말고사</th> <th style="width: 12.5%;">퀴즈</th> <th style="width: 12.5%;">발표</th> <th style="width: 12.5%;">프로젝트</th> <th style="width: 12.5%;">과제물</th> <th style="width: 12.5%;">참여도</th> <th style="width: 12.5%;">기타</th> </tr> </thead> <tbody> <tr> <td>30 %</td> <td>30%</td> <td>%</td> <td>%</td> <td>30%</td> <td>10%</td> <td>0%</td> <td>%</td> </tr> </tbody> </table>	중간고사	기말고사	퀴즈	발표	프로젝트	과제물	참여도	기타	30 %	30%	%	%	30%	10%	0%	%
중간고사	기말고사	퀴즈	발표	프로젝트	과제물	참여도	기타									
30 %	30%	%	%	30%	10%	0%	%									

## II. 교과목표(Course Objectives)

본 과목에서는 **GPU 상에서의 CUDA 기반 병렬 프로그래밍 능력을 습득**하기 위하여 다음을 비롯한 관련 주제를 학습한다.

1. 프로그래머 관점에서의 GPU architecture
2. CUDA 시스템의 이해: Programming model, execution model, SIMT processing, concurrency 등
3. C++를 확장한 CUDA API 및 kernel 함수 기반의 병렬 프로그래밍 기법
4. 기본적인 SIMT 기반 병렬 처리 기법
5. 응용 문제로의 적용

상기 주제에 대한 학습을 통하여, 어떠한 병렬성을 가진 문제에 대하여 GPU가 효과적으로 활용될 수 있는지 이해하고, 그러한 문제를 GPU를 사용하여 효과적으로 해결할 수 있는 문제 해결 능력을 습득한다.

## III. 수업운영방식(Course Format)

(\* I-3의 수업방법의 구체적 설명)

- 정규 수업시간에 GPU 프로그래밍 관련 이론과 CUDA 프로그래밍 기법 전반에 대한 강의를 진행함.
- 이와 함께, 각 주제와 관련한 예제 코드를 다룬 후, 그에 따른 프로그래밍 숙제를 부여할 예정임.
- 따라서 **Windows와 Visual Studio를 기반으로 NVIDIA GPU에서 CUDA 프로그래밍을 할 수 있는 환경(개인적으로나 또는 학과 실습실을 통하여)을 각자 구축해야 함. 구축에 어려움이 있는 학생은 담당 교수와 면담을 할 것.**  
**(AMD GPU No! MAC PC No! OpenCL No! Python No! Only C/C++ with CUDA!)**

## IV. 학습 및 평가활동(Course Requirements and Grading Criteria)

- 중간고사 및 기말고사 등 두 번에 걸쳐 필기시험을 본다.
- 필요할 경우 적절하게 수업시간에 퀴즈를 보거나 문제 풀이 숙제를 부여한다.
- 몇 차례에 걸쳐 프로그래밍 숙제를 부여한다.

[참고] "1-4"의 성적 평가방법은 대략적이며, 수업의 진행 상황에 따라 (적은) 범위의 변경이 있을 수 있음.

## V. 수업규정(Course Policies)

- 수업 진행과 관련하여 학칙과 보편적인 관례를 따른다.
- 수업 중 타인에게 피해가 되거나 수업 분위기를 해치는 행동은 금지한다.
- C/C++ 기반의 CUDA 프로그래밍 숙제는 혼자서 하는 것이 원칙으로 함.
- **청강은 허용하지 않는다.**

## VI. 교재 및 참고문헌(Materials and References)

- 본 과목 제공 PPT 자료
- 본 과목 제공 기술 자료
- 본 과목 제공 예제 코드
  
- R. Ansgore, Programming in Parallel with CUDA: A Practical Guide, Cambridge University Press, 2022.
- W. Hwu et al., Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann, 2022.
- J. Cheng et al., Professional CUDA C Programming, John Wiley and Sons, 2014.
  
- NVIDIA, CUDA C++ Programming Guide(Release 12.3), November 2023 또는 이후 버전.
- NVIDIA, CUDA Runtime API: API Reference Manual, January 2023 또는 이후 버전.
- 기타 NVIDIA 기술 문서

## VII. 주차별 강의계획(Course Schedule)

[참고] 본 과목에서 다룬 내용들은 서로 복잡하게 연관되어 있어 아래 강의계획의 내용과 순서는 강의를 진행하면서 적절히 변경할 예정이다.

1 주차	학습목표	<b>Introduction to GPU Architecture</b>
	주요학습내용	본 과목에 대한 소개 후, 최신 NVIDIA GPU의 구조에 대하여 이해한다.
	수업방법	강의
	수업자료	<b>VI. 교재 및 참고문헌 참조</b>
	과제	추후 결정
2 주차	학습목표	<b>Introduction to CUDA System and Execution Model</b>
	주요학습내용	CUDA 시스템 전반에 대한 소개 후, CUDA의 SIMT 기반 계산 구조를 배운다.
	수업방법	강의
	수업자료	<b>VI. 교재 및 참고문헌 참조</b>
	과제	추후 결정
3 주차	학습목표	<b>CUDA Programming Model</b>
	주요학습내용	Kernel과 thread hierarchy 개념에 대하여 이해한 후, CUDA Runtime API와 CUDA C++ Extensions을 사용한 명령 처리 및 프로그래밍 방식을 배운다.
	수업방법	강의
	수업자료	<b>VI. 교재 및 참고문헌 참조</b>
	과제	추후 결정
4 주차	학습목표	<b>CUDA Memory Hierarchy</b>
	주요학습내용	Register, L1/L2 cache, shared memory, global memory, texture memory, 그리고 constant memory 등 각 메모리의 구조 및 특성에 대하여 이해한다.
	수업방법	강의
	수업자료	<b>VI. 교재 및 참고문헌 참조</b>
	과제	추후 결정
5 주차	학습목표	<b>Efficiency in CUDA Programming</b>
	주요학습내용	Occupancy, divergence, global memory coalescing, shared memory bank conflict 등과 같은 성능에 영향을 미치는 요소들을 예제 코드를 통하여 배운다.
	수업방법	강의
	수업자료	<b>VI. 교재 및 참고문헌 참조</b>
	과제	추후 결정
6 주차	학습목표	<b>Matrix Multiplication and Image Filtering</b>
	주요학습내용	행렬의 곱셈과 2D 영상 처리 문제를 예로 하여 효율적인 CUDA 프로그래밍 기법을 익힌다.
	수업방법	강의
	수업자료	<b>VI. 교재 및 참고문헌 참조</b>
	과제	추후 결정
7 주차	학습목표	<b>Concurrency Using CUDA Streams and Events I</b>
	주요학습내용	CUDA가 제공하는 Stream과 Event 기능을 사용한 프로그래밍 기법을 배운다.
	수업방법	강의
	수업자료	<b>VI. 교재 및 참고문헌 참조</b>
	과제	추후 결정
8 주차	학습목표	평가
	주요학습내용	한 학기 동안 배운 내용에 대하여 평가한다.
	수업방법	중간고사 시험
	수업자료	
	과제	

9 주차	학습목표	<b>Concurrency Using CUDA Streams and Events II</b>
	주요학습내용	CPU와 GPU가 제공하는 리소스를 최적으로 활용할 수 있는 CUDA의 비동기적 병렬 처리 모델을 이해한 후, 예제 코드를 통하여 실험하여 본다.
	수업방법	강의/튜토리얼
	수업자료	VI. 교재 및 참고문헌 참조
	과제	추후 결정
10 주차	학습목표	<b>Cooperative Groups and Dynamic Parallelism</b>
	주요학습내용	전통적인 32-thread warp 개념을 확장한 cooperative groups 기능과 동적 커널 함수 호출을 위한 dynamic parallelism 기능에 대하여 배운다.
	수업방법	강의
	수업자료	VI. 교재 및 참고문헌 참조
	과제	추후 결정
11 주차	학습목표	<b>Warp-Level Primitives and Atomic Operations</b>
	주요학습내용	CUDA의 warp-level primitive와 atomic operation을 사용한 프로그래밍 기법을 배운다.
	수업방법	강의
	수업자료	VI. 교재 및 참고문헌 참조
	과제	추후 결정
12 주차	학습목표	<b>Parallel Reduction, Prefix Sum, and Compaction</b>
	주요학습내용	병렬 처리 분야의 기본 알고리즘들인 reduction, prefix sum, 그리고 compaction 등을 이해한 후, CUDA를 사용한 이들의 효율적인 구현 기법을 습득한다.
	수업방법	강의
	수업자료	VI. 교재 및 참고문헌 참조
	과제	추후 결정
13 주차	학습목표	<b>Tensor Cores</b>
	주요학습내용	인공지능 계산에서 요구하는 행렬 곱셈을 가속하기 위한 NVIDIA GPU의 Tensor Cores 기능을 활용하는 방식을 배운다.
	수업방법	VI. 교재 및 참고문헌 참조
	수업자료	강의
	과제	추후 결정
14 주차	학습목표	<b>고급 GPU 프로그래밍 기법의 이해 I</b>
	주요학습내용	컴퓨터 그래픽스, 영상 처리 등 여러 분야에서 널리 사용되는 Marching Cubes 알고리즘을 구현한 NVIDIA 제공 예제 코드의 분석을 통하여 실제 응용문제 해결을 위한 고급 CUDA 프로그래밍 기법을 배운다.
	수업방법	강의
	수업자료	VI. 교재 및 참고문헌 참조
	과제	추후 결정
15 주차	학습목표	<b>고급 GPU 프로그래밍 기법의 이해 II</b>
	주요학습내용	컴퓨터 그래픽스, 영상 처리 등 여러 분야에서 널리 사용되는 Marching Cubes 알고리즘을 구현한 NVIDIA 제공 예제 코드의 분석을 통하여 실제 응용문제 해결을 위한 고급 CUDA 프로그래밍 기법을 배운다.
	수업방법	강의
	수업자료	VI. 교재 및 참고문헌 참조
	과제	추후 결정
16주차	학습목표	평가
	주요학습내용	한 학기 동안 배운 내용에 대하여 평가한다.
	수업방법	기말고사 시험
	수업자료	
	과제	

## VIII. 참고사항(Special Accommodations)

- 장애로 인하여 수강 시 지원이 필요한 학생들은 개별적으로 상의하기 바랍니다.