Module title Graphics Processing Unit (GPU) Computing				
Module code YGPU	**Level** Bachelor (B.Sc.) IN, IT	**Hours per week** 4	**ECTS credits** 5	**Duration** 2-3 weeks

Module instructor	**Lecture type**	**Prerequisite(s)**	**Grading**
Mr. Lee Wai Kong, University Tunku Abdul Rahman, Malaysia	Interactive seminar Individual consultations	Good academic standing	Exam Mini Project Practical Assessment

Objectives

- To provide an understanding in the aspects of hardware, software, programming environment and performance profiling for general purpose computing in GPU.
- To develop the knowledge and skills for designing parallel processing applications using GPU.
- To study the techniques for optimizing parallel algorithms in GPU platform.

Content

This subject introduces the concepts, languages, techniques, and patterns for general purpose GPU computing. GPU can be used as massively parallel co-processor to parallelize many serial algorithms as well as accelerate existing parallel algorithms. It covers GPU architectures, data-parallel programming models, techniques for memory bandwidth optimization and parallel algorithm patterns. The students will learn the techniques to develop parallel applications in GPU platform and evaluate its performance.

- Topic 1: Introduction to parallel programming platforms and system architectures
  Flynn's Taxonomy; Homogeneous (CPU) and Heterogeneous (CPU + GPU); computing system; Vertical scaling vs. Horizontal scaling; Introduction to Parallel programming languages (CUDA, OpenMP and OpenCL).
- Topic 2: Introduction to basic parallel programming concepts
  Sequential programming vs. parallel programming paradigms; Identifying overheads and bottleneck of sequential application.; Data sharing and synchronization; Well known parallel solutions such as partitioning, and divide-and-conquer; Techniques to identify concurrency opportunities.
- Topic 3: GPU Architecture and Programming Model
  Introduction to GPU memory model in GPU (global, shared, register, constant and texture memory); Programming model for GPU: Single Instruction Multiple Data (SIMD); Grid, blocks and thread blocks; Introduction to GPU programming language.
- Topic 4: Performance Metrics for Parallel Systems
  Parallel performance metrics (total overhead, speedup, efficiency); Amdahl's Law vs. Gustafson's Law; Parallel Overhead; Profiling tools for GPU computing.
- Topic 5: GPU Memory Model
  Common techniques for parallelizing serial code in GPU; Global memory bandwidth (coalesced memory access pattern); Shared memory and bank conflict; Constant and texture memory; Register spilling and local memory.
- Topic 6: Optimization Techniques
  Identifying bottleneck for parallel program (memory bound or compute bound); Concurrent execution of CPU program, GPU kernel and memory copy process; Thread blocks ordering; Occupancy; Stream programming model.

- Topic 7: Mini Project - The students will be given a list of algorithms to choose for parallel implementation. The students need to implement and optimize the selected algorithms using GPU. Example algorithms: Encryption: AES, IDEA, Threefish; Hash Function: BLAKE, Keccak, SHA-1, SHA-2; Public Key Cryptography (Montgomery Multiplication, Karatsuba Multiplication); KNN; Binary Tree, Red Black Tree; Pseudorandom Number Generator; Matrix Solver (Dense or Sparse, Direct or Iterative); Various Search and Sort algorithms; Etc.

Textbook/teaching material

- Lecturer provided materials on e-learning platform.

Note: this is not the official course descriptor according to the "Studien- und Prüfungsordnung" (SPO)