Parallel Programming

1. **Course number and name**: 020PPLES5 Parallel Programming

2. **Credits and contact hours**: 4 ECTS credits, 2x1:15 contact hours (course + lab)

3. **Name(s) of instructor(s) or course coordinator(s)**: Maroun Chamoun

4. **Instructional materials**: Handouts posted on the Web

5. **Specific course information**
   a. **Catalog description**:
      Parallel architectures – Parallel Computing – Concurrency and Threads – Parallelism in C++ 17 & OpenMP – Message Passing Interface (MPI) – Heterogenous programming and GPUs.

   b. **Prerequisites*:*** 020CPPES1 Object-Oriented Programming

   c. **Required** for CCE Software Engineering option students; **Selected Elective** for CCE Telecommunication Networks option students

6. **Educational objectives for the course**
   The goal of this course is to provide an in-depth introduction to parallel programming libraries and APIs in C++ and application of those libraries to solve complex problems.

   a. **Specific outcomes of instruction:**
      - Gain knowledge of parallel architectures.
      - Optimize the system resources of the computer system.
      - Parallelize a mathematical model, design a solution and develop the code using different parallelism models (Shared memory model, distributed systems, heterogeneous CPU-GPU programming).

   b. **PI addressed by the course:**

PI	1.1	1.2	1.3	2.1	2.2	2.4	6.4	7.1
Covered	x	x	x	x	x	x	x	x
Assessed			x					

7. **Brief list of topics to be covered**
   - Parallel Architectures: Flynn's classification - Single-core architectures - Multi-core architectures - Shared memory parallel architecture - Distributed memory parallel architecture (3 lectures)
   - Parallel Computing: Data parallelism - Control parallelism - Stream parallelism - The four phases of Foster's PCAM approach (1 lecture)
     Tutorials: Illustration of the PCAM approach – Parallelization of operations on polynomials (1 lecture)

Limits of parallelism - Performance indicators - Amdahl's law - Gustafson's law (1 lecture)
Tutorials: Evaluation of the performance of the two-step calculation on an N x N image (1 lecture)

- Concurrency and Threads: Concurrency vs parallelism - Threads vs Processes - Threads in C++ - Atomic execution in C++ - Synchronization with Mutex - Synchronization with condition variables - Synchronization with Future/Promise - Asynchronous execution with async (1 lecture)
  Tutorials: The barbershop (use of Mutex) - The barbershop (using Condition Variables) (1 lecture)
  Tutorial: The Barber Shop (using Future/Promise) (1 lecture)
  Lab: Creating and Using Semaphores in C++ for Synchronization and Mutual Exclusion (1 lecture)
- Parallelism in C++17 & OpenMP: Parallel algorithms in C++17 (1 lecture)
  OpenMP: Execution Mode - Memory Model - Directives - Subroutine - Parallel Region - Shared and Private Variables - Parallel Loops - Reduction - Conditional Parallelization - Parallel Sections - Exclusive Execution - Synchronization - Critical Section (2 lectures)
  Tutorial: Vector product using OpenMP (1 lecture)
  Lab: Using OpenMP to develop parallel programs - Ensuring synchronization using the barrier directive - Ensuring mutual exclusion using the critical directive (1 lecture)
- Message Passing Model (MPI): Passing messages in distributed memory - Model by message exchange - MPI programming model (Message Passing Interface) - Structure of an MPI program - Management of communications - Initialization of the MPI environment – Identifier - Point-to-point communication - Collective Communications - Derived Data Type (3 lectures)
  Tutorials: Calculation of the scalar product - Multiplication of a matrix by a vector - Multiplication of two matrices (1 lecture)
  Lab: Developing a parallel application using the messaging model (1 lecture)
- Parallel Programming with GPUs: Architecture of GPUs - CUDA programming model - Operation of a hybrid program - Variables and CPU/GPU data transfers - Structure of the Host part - Definition and execution of a kernel - Multidimensional CUDA grids - Sharing data between d 'a same block - Optimizations: Memory throughput, Limitation of divergence, streams Compilation of a CUDA application (3 lectures)
  Tutorials: Calculation of the scalar product (1 lecture)
  Lab: Multiplication of a matrix by a vector (1 lecture)
- Vector units: Scalar calculation vs vector calculation - General operation of SIMD units - The AVX unit & the intrinsics (2 lectures)