Architecture and Hardware (COMP0068)

**Description**

**Aims:**
The aim of this module is to provide fundamental knowledge about how a modern computer works. We will provide an overview of the key hardware components making up a computer and how these components work together. We will examine the architecture of a real processor including its assembly language and machine code. We will learn about the fundamental aspects of operating systems and how these provide an interface between the user (application program) and the computer hardware. We will learn how high-level languages can be translated into assembly language and then low-level machine code which drives the computer hardware. Students will become familiar with computer terminology, calculating performance metrics and key design aspects which make modern computers so efficient.

**Learning outcomes:**
- Manipulate different computer data representations (binary, hex, floating point numbers, machine code instructions, etc.);
- Describe the internal structure of a processor and how it works;
- Explain how modern computers handle memory and input/output;
- Write and understand assembly language programs;
- Describe how high-level languages are translated to assembly language and subsequently machine code that drives the hardware;
- Describe how operating systems provide an interface between the user / application program and the underlying hardware of the computer;
- Explain how key aspects of the operating system work such as handling processes, threads, file systems and memory management;
- Calculate the computer performance of different hardware devices and understand key design aspects that have made modern computers so efficient;
- Describe in detail how, overall, a modern computer works

**Key information**

- **Year:** 2019/20
- **Credit value:** 15 (150 study hours)
- **Delivery:** PGT L7, Campus-based
- **Reading List:** [View on UCL website](ucl.ac.uk)
- **Tutor:** Dr Ghita Kouadri Mostefaoui
- **Term:** Term 1
- **Timetable:** [View on UCL website](ucl.ac.uk)

**Assessment**

- Written examination (main exam period): 90%
- Coursework: 10%

**Find out more**

For more information about the department, programmes, relevant open days and to browse other modules, visit [ucl.ac.uk](ucl.ac.uk)

**Disclaimer:** All information correct as of August 2019. Please note that aspects of the module may be subject to change. UCL will make best efforts to inform applicants of major changes.
including accurately employing key computer
terminology and reasoning about computer
performance based on it.

Content:
- Top-down high level overview of a computer: the
  main components making up a computer and how
  they interact;
- Computer arithmetic: binary, hexadecimal, fixed
  size arithmetic, 2s-complement, 16s-complement;
- Data representation: big-endian, little-endian,
  fixed point, floating point, character strings;
- Assembly language and machine code;
- Translation of high-level languages into assembly
  and object/executable files;
- Inside the CPU: internal structure of a processor
  that allows it to execute machine code;
- Memory and disk: understanding basic concepts
  about primary and secondary storage, and
  learning their performance measures;
- Input/Output: memory-mapped I/O, software
  polling, interrupt mechanisms and direct memory
  access (DMA);
- Structure of operating systems: processes and
  process scheduling, threads, synchronization,
  memory management and file systems;

Requisites:
In order to be eligible to select this module, a
student must be registered on a programme for
which it is a formally-approved option or elective
choice AND have A-level Mathematics (or
equivalent).