# COMP 201 FUPE Study Guide

## Format

The exam consists of several pages of problems (you will provide your own paper for the answers). Passing is 80%. It is two hours in length and no additional materials – such as book or notes – are permitted. For those students where English is a second language, a translation dictionary may be used.

## Notes

This is a difficult exam as it is all problems, short answer, and essay questions (no multiple choice, fill in, or true/false). You will be expected to do problems and answer questions related to number conversions, differing base arithmetic, fetch/execute cycle, pipelining and superscalar execution, assembly language programming, instruction set architectures, and microprogramming.

## Recommended textbook

Englander, I. (2009). *The architecture of computer hardware, systems software, and networking: An information technology approach*. (4th). Hoboken, NJ: Wiley. ISBN: 978-0-471-71542-9. . The textbook publisher has a companion website with useful information.

## Course Description

This course is the first of four courses that holistically explore the structure of computational systems. This course deals with the nature of computer hardware. The course will cover the structure of current computer systems at the level of functional organization, representation of data and programs, the design of the memory hierarchy, and, the design of the I/O system. The course will introduce basic assembly language.

## Major topics

Topics included in this course are:

- Introduction to computer architecture
- Number systems and data presentation
- Design principles
- The Little Man Computer model
- Programming in assembly language
- Performance measures
- Memory hierarchy
- Performance enhancement techniques
- Input/Output
- Peripherals and networking

# Course Outcomes

1. Analyze the structure of computer systems.
2. Analyze the relationship between computer system structures and performance.
3. Develop assembly language programming skills.
4. Demonstrate the implementation of higher-level language constructs in assembly language.

# Module Outcomes

## Module 1 (1 week)

1. Explain basic computer architecture and why it is important.
2. Trace the historical development of microprocessors including the Intel family.
3. Compare different number systems by converting numbers and performing arithmetic in each system, as specified.

## Module 2 (1 week)

1. Demonstrate the representation of numbers and texts using various formats.
2. Perform conversion and arithmetic operations on integer and floating point formats.

## Module 3 (5 weeks)

1. Describe the Little Man Computer model and write assembly language programs to solve problems with it.
2. Describe an Instruction Set Architecture and differentiate between RISC and CISC implementation.
3. Describe the fetch-execute cycle and pipelining.
4. Use basic measures of computer performance and describe why some can be misleading.
5. Describe memory hierarchy and techniques to enhance the performance.
6. Explain the structure and performance characteristics of magnetic disks.
7. Describe common computer peripheral devices and computer networking.