

A Proposal for a New Second Year Course:  
**Introduction to Computer Engineering**

Larry Hughes, PhD  
Electrical and Computer Engineering, Dalhousie University  
14 August 2017

## **1 Overview**

Many students opting for Dalhousie Computer Engineering have an interest but limited knowledge of computer systems in general and computer engineering in particular. Their body-of-knowledge of computer systems and computer engineering is poor compared to that of Computer Engineering students completing their second year at other universities. The course, *Introduction to Computer Engineering*, is intended to address this.

The course has two components:

**Design.** The design component of the course focusses on software design, implementation, testing, and documentation. The importance of both the design and testing of software is emphasized through the development of a modular solution to a term-long project.

**Concepts.** The concepts component is an overview of different areas of computer engineering. It is intended to give the student an understanding of existing and possible future computer systems and applications, the need for life-long learning, and possible career paths.

The course is based on the course *ECE 101: Introduction to ECE*, described in the sample Computer Engineering program developed by ACM-IEEE for Electrical and Computer Engineering departments (ACM-IEEE, 2016).

## **2 Outcomes**

At its conclusion, students completing this course should demonstrate:

1. A measurable improvement in their ability to take a problem, design a software solution, implement the solution in a programming language, apply a set of tests to demonstrate the correctness of their solution, and write the necessary documentation for the solution.
2. An understanding of the history of computers and computer engineering, the current state-of-the-art, and future of computers and computer engineering.

## **3 Course description**

This course introduces second year Computer Engineering to software design, implementation, testing, and documentation. The course also examines selected topics in Computer Engineering. A multi-part term project applies concepts to a software solution to a problem.

## **4 Course contents**

This course consists of two components: the design, implementation, and testing of a software solution to an application; and an introduction to Computer Engineering concepts. About 80%

of the course should be devoted to the design component. The design component can be approached in a top-down, bottom-up, or mixed fashion.

The project should be divided into multiple parts, corresponding to each part of the course, starting with the design concepts.

To give the students additional opportunity to practice their software design, implementation, and testing skills, the labs can be *programming labs*, in which the students are to develop a solution for a given problem. A possible source of these problems are past ACM Programming Contests.

Suggested contents:

1. Introduction to course
2. Basic software design concepts
3. Implementing the design
4. Testing techniques
5. The user interface
6. Writing technical documentation
7. Software maintenance
8. Course review
9. Final examination

## **5 Concepts component topics**

The Concepts component can include topics such as:

- Computer Engineering
- Computer Systems
- File Systems
- Databases
- Operating Systems
- Programming Languages
- Programming style
- Communications
- Computer Architecture
- Computer graphics and imaging
- Applications
- Others, such as Artificial intelligence, Computer security, Embedded Systems, IoT (Internet of Things)

This part of the course can be team-taught with specialized topics taught by, for example, departmental members, academics from other faculties, local and regional experts, and on-line lectures.