

Fall 2019

CIS-261 Home: <http://www.c-jump.com/bcc/c261c/CIS261syllabus.html>

**CIS-261 Introduction to Computer Systems
Bristol Community College
Computer Information Systems Department**

Catalog Description

This course is an introduction to major components of computer systems. The course introduces fundamental concepts of computing systems such as binary arithmetic and data representation, the Von Neumann model for processing computer programs, the operation of memory, instruction set, and machine and assembly language programming. It systematically presents the levels of transformations from machine language to assembly language to high-level language. The course studies the role of such systems software components as assemblers, compilers, linkers, loaders, and operating systems. The course has a strong project component. Pre or co-requisite: CIS 158 or permission of the instructor. Co-requisite: MTH 243. Three lecture hours and two lab hours per week. Instructional Support Fee applies. Competency met: Technical Literacy (8.0) 4 credits.

Pre- and co-requisites:

Pre-requisites: CIS 158 (formerly CIS 76) or permission of the instructor.
Co-requisite: MTH 243 (formerly MTH 43) or permission of the instructor.

Instructor: Igor Kholodov Igor.Kholodov@bristolcc.edu
Office: K211
Telephone: 774-357-3328

Meeting days and times:

Lecture: room K-105 Tuesdays 8:00 am-10:40 am.
Laboratory: room K-105 Thursdays 9:30 am-10:45 am..

Student Learning Outcomes

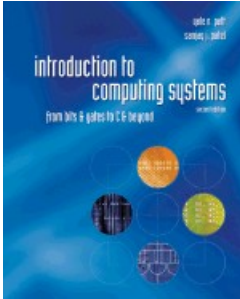
1. Identify and describe different types of computing machines.
2. Describe distinguishing components of Von-Neumann Architecture.
3. Understand principals of machine language and program in assembly.
4. Describe the relation between high level languages and assembly.
5. Explain how bits and bytes are organized to represent data electronically.
6. Describe how digital logic is used to build circuits to process information.

Course Objectives

Upon completion of this course, the students should be able to do the following:

1. Describe how computing machines are organized internally
2. Identify and explain the design trade offs between different computing machines
3. Describe how computers perform calculations
4. Define systematic sequence of transformations between layers of abstraction.
5. Indicate program organization in memory
6. Understand program stack, stack frames, procedure calls, and parameters

Recommended Text



Authors: Sanjay J. Patel, Yale Patt

Title: Introduction to Computing Systems: From Bits and Gates to C and Beyond.

Publisher: McGraw-Hill, 2nd Edition, 2004.

ISBN-13: 9780072467505

ISBN-10: 0072467509

Course Outline

1. Number systems, Bits, Data Types, and Operations.
 - Presentation: Introduction to Computer Systems
 - Presentation: Bits, Numbers, and Binary Data Representation
 - **Assignment A01**: Pages 17-20, Exercises 1.4, 1.9, 1.10, 1.15-17, 1.21 and 1.23
 - Reading: Textbook Chapters 1 and 2.
2. Digital Logic Structures
 - Presentation: Digital Logic
 - Presentation: Arithmetic Operations and Logical Functions
 - **Lab L01**: Pages 43-50, Exercises 2.10-11, 2.13-14, 2.17 (interpret bits as signed numbers, use calculator, sign-extend to 8 or 16 bit quantities as necessary), 2.18-19, 2.30, 2.32-35, 2.43 and 2.45-50 (use calculator as necessary), 2.53-54
 - Reading: Textbook Chapter 3.
3. Memory and Sequential Logic Circuits
 - Presentation: Storage Elements
 - Presentation: Sequential Logic Circuits
 - **Lab L02**: Using Digital Works
 - Download Digital Works
 - **Assignment A02**: Pages 43-50, Exercises 2.1-3, 2.5, 2.8, 2.12, 2.15, 2.21 (hint: refer to the "Overflow Condition" described in first week's lecture), 2.23, 2.27-29, and 2.31
 - Reading: Textbook Chapter 4.
4. The Von Neumann Model
 - Presentation: The von Neumann Computer Model
 - **Lab L03**: Two's Complement Adder/Subtractor

- **Assignment A03:** Pages 82-91, Exercises 3.1, 3.10, 3.13-14, 3.23, and 3.32
 - Reading: Intel Basic Architecture Volume 1 Chapter 2, **Introduction to the Intel Architecture.**
5. Introduction to x86 Assembly Language
- Presentation: Introduction to x86 Assembly Language
 - Presentation: The Assembly Process
 - Discussion: Overflow Detection Circuits
 - **Lab L04:** Overflow Detector
 - **Assignment A04:** Pages 111-114, Exercises 4.1-4, 4.5a, 4.5b.1, 4.5b.2, 4.5b.4, and 4.9.
 - Reading: Intel Basic Architecture Volume 1 Chapter 3, **Basic Execution Environment.**
6. Instruction Set Architecture Design
- Presentation: Instruction Format Design
 - Presentation: Encoding Real x86 Instructions
 - Presentation: The Instruction Cycle
 - Quick Reference: x86 Instructions by Opcode
 - Quick Reference: x86 Instructions by Mnemonic
 - Quick Reference: Brief x86 Instruction Set Reference
 - **Lab L05:** Design of a 4-bit Magnitude Comparator
 - **Assignment A05:**
 - Install Visual Studio.
 - Use Visual C++ Development Environment to assemble and debug a do-nothing program `M1.ASM`
 - Reading: Intel Instruction Set Reference Volume 2 Chapter 2, **Instruction Format.**
7. Language Components of MASM
- Presentation: Language Components of MASM
 - Registers, EFLAGS, Addressing Modes, Effective Address.
 - **Lab L06:** Boolean Algebra Theorems and Sum of Products
 - Reading: Intel Instruction Set Reference Volume 2 Appendix B, **Instruction formats and encodings.**
8. Data Types and Memory Allocation
- Presentation: Data Types and Memory Allocation
 - Endianness, Integer Formats, Copying Data.
 - Reference: Intel Volume 1, **Basic Architecture.**
 - Reference: Intel Volume 2, **Instruction Set Reference.**

- **Lab M07:** Opcode Analysis using OllyDbg
9. CPU Flags and Data Manipulation
- Presentation: CPU Flags and Data Manipulation
 - ADD, INC, SUB, DEC, NEG, Carry Flag, Overflow Flag, SHL, SHR, and SAR
 - Quick Reference: Intel Basic Architecture Volume 1
 - Appendix A: **EFLAGS cross-reference**
 - Appendix B: **EFLAGS condition codes**
 - **Lab M08:** Console Input/Output
10. Controlling Program Flow
- Presentation: Controlling Program Flow
 - **Lab M09:** Data Arrays, Windows API Calls
11. Modes of Memory Addressing on x86 and Logical Segments
- Presentation: Modes of Memory Addressing on x86
 - Reading: Intel System Programming Guide Volume 3 Chapter 3, **Protected-mode memory management.**
 - **Lab M10:** Integer Arithmetic Part 1: High-speed Multiplication by Powers of 2
12. Memory Protection, Paging, and Memory Model Directives
- Presentation: Memory Protection
 - Presentation: Logical Segments and Memory Model Directives
 - Reading: Intel System Programming Guide Volume 3 Chapter 4, **Memory protection.**
 - **Lab M11:** Integer Arithmetic Part 2: Extended Precision Techniques
13. Instruction Operand Addressing on x86
- Presentation: Instruction Operand Addressing on x86
 - Reading: Intel Basic Architecture Volume 1 Chapter 5, **Data Types and Addressing Modes.**
 - **Lab M12:** Step-by-step execution analysis
14. The Program Stack and Procedures
- Presentation: The Program Stack
 - Presentation: Procedures
 - Reading: Intel Basic Architecture Volume 1 Chapter 4, **Procedure Calls, Interrupts, and Exceptions.**
 - **Lab M13:** Building Your Own Stack Machine **Final Project**
15. Final Project Review

Reference Materials

Reference: Intel Volume 1, **Basic Architecture**
Reference: Intel Volume 2, **Instruction Set Reference**
Quick Reference: x86 Instructions by Opcode
Quick Reference: x86 Instructions by Mnemonic
Quick Reference: Brief x86 Instruction Set Reference
OllyDbg Debugger for Microsoft Windows: a debugger that emphasizes *binary code analysis*.

Credit Hour Workload

Student Success Tip: *For each hour in class, you should expect to study at least 2-3 hours outside of class. Know your limits, avoid over scheduling yourself (whether it be work or class). Set up a schedule that you know will allow you to earn good grades. And, maintain a day planner to help you stay organized.*

Disability Accommodations

If you are a student who would normally seek accommodations in a traditional, face to face classroom, or if you have a documented disability and will be requiring accommodations, please contact me and the Office of Disability Services (508) 678-2811 (Fall River, ext. 2955; Attleboro and Taunton, ext. 2996, New Bedford, ext. 4011) as soon as possible to arrange for appropriate accommodations. You may also contact Office of Disability Services (ODS) online at <http://www.bristolcc.edu/student-services/resources/disability-services/>

Minimum Requirements for a Passing Grade

Curiosity and love for learning
Ability to work with others in thinking and learning together
Ability to do online research for any subject that is not clear in text or lecture handouts. Completion of 80 percent of all homework assignments.
Completion of 80 percent of all lab assignments.
A grade of 60 or greater on the final project.

Weights for the Final Grade Determination

Homework Assignments	30%
Lab Assignments	50%
Final Project	20%

The Final Grades will be assigned as follows:

97 - 100 A+	87 - 89 B+	77 - 79 C+	67 - 69 D+
93 - 96 A	83 - 86 B	73 - 76 C	63 - 66 D
90 - 92 A-	80 - 82 B-	70 - 72 C-	60 - 62 D-
			Below 60 F

Teaching Methodology

The class web page samples, the study guide that references the textbook, online quizzes, and the homework assignments are the principal teaching methods that will be used in this course. "Handouts" and other instructor samples will be available on the class web page.

Bristol Academic Policies

College-wide Academic Policies outlined in Academic Catalog directly apply to this course. It is your responsibility to read carefully and understand Academic Information, especially Academic Integrity, Academic Dishonesty, Academic Negligence, Plagiarism, and Classroom conduct, published online. See Academic Catalog for details: <http://bristolcc.edu/>

Attendance Policy

considered "present" for the week if they submit the required lab assignment (with a satisfying passing grade) prior to the due date for that week. Poor attendance may affect your final grade.

Attendance is recorded weekly based on the student's ability to submit quality and timely lab/programming assignments each week. Students are responsible for withdrawing officially if they stop attending any or all classes. Faculty no longer have the ability to withdraw a student from a class. A grade of "F" will be assigned to any student who stops attending a course but does not officially withdraw. Students are encouraged to meet with an advisor before making any changes to their schedule. Withdrawals impact Satisfactory Academic Progress and can place the student at risk for academic probation or dismissal. Students who use financial aid and who subsequently withdraw may be required to return some or all funds received. Withdrawals are accepted until the tenth week of classes. Students may withdraw online in accessBCC, in person at any Enrollment Center, or via their college email to enrollmentservices@bristolcc.edu. Email requests must come from the student's college email address and must include the student's name, student ID number, and course information (CRN, course and section number).

Email from non-college accounts will not be accepted. If a student officially withdraws after the third week of classes, there will be no tuition or college fee refunds. For more information, see the College Catalog at

<http://bristolcc.edu/>

Students with questions should contact Enrollment Services via any of the methods mentioned above or at 774-357-2590.

NEED HELP OR SUPPORT?

If you feel overwhelmed and need help or support at any time, please know that Bristol faculty & staff are here for you and we can address your needs or refer you to the appropriate resources necessary! Contact us to make a referral to Bristol's Early Alert/CARE Team that can provide many resources to you that are available. Your well-being and success is our top priority. Find out more about EarlyAlert/CARE Team:

<http://www.bristolcc.edu/bristolcommunity/facultystaff/careteam/>



This syllabus is based upon work supported by the [National Science Foundation](#) under Grant No. 0122636. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

Note: *This syllabus is a suggested course outline and will be generally followed, subject to change according to the instructor's discretion and needs. Academic flexibility is important.*

@inproceedings{Patt1999IntroductionTC, title={Introduction to computing systems - from bits and gates to C and beyond (2. ed.)}, author={Yale N. Patt and Sanjay J. Patel}, year={1999} }.Â " This text covers (in order) a switch level abstraction of a MOS Transistor, Logic Gates, latches, CONTINUE READING. View PDF. Save to Library. Create Alert. Cite. 630 p. ISBN-10 0072467509; ISBN-13 978-0072467505. Introduction to Computing Systems: From bits & gates to C & beyond, now in its second edition, is designed to give students a better understanding of computing early in their college careers in order to give them a stronger foundation for later courses. The book is in two parts: (a) the underlying structure of a computer, and (b) programming in a high level language and programming methodology. To understand the computer, the authors introduce the LC-3 and provide the LC-3 Simulator to give students hands-on access for testing what Introduction to Computing Systems: From Bits and Gates to C and Beyond. 2nd Edition. Yale N. Patt Sanjay J. Patel. Original slides from Gregory Byrd, North Carolina State University Modified by Chris Wilcox, S. Rajopadhye Colorado State University. CS270 Fall 2013 - Colorado State University. 1. Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display. Lecture Goals. ! Review course logistics. nn Assignments nn Policies nn Organization nn Grading Criteria.

630 p. ISBN-10 0072467509; ISBN-13 978-0072467505. Introduction to Computing Systems: From bits & gates to C & beyond, now in its second edition, is designed to give students a better understanding of computing early in their college careers in order to give them a stronger foundation for later courses. The book is in two parts: (a) the underlying structure of a computer, and (b) programming in a high level language and programming methodology. To understand the computer, the authors introduce the LC-3 and provide the LC-3 Simulator to give students hands-on access for testing what Yeah, reviewing a book introduction to computing systems from bits gates to c beyond could be credited with your near connections listings. ons.oceaneering.com. ons.oceaneering.com/.../viewcontent.php? introduction.to.computing . systems.from.bits . gates.to.c . beyond clipped from Google - 10/2020. Gratis Introduction To Computing Systems: From Bits & Gates To C infoelbasani.al. The third edition of Introduction to Computing Systems: From bits & gates to C / C++ and beyond is designed to give students a strong foundation infoelbasani.al. https://infoelbasani.al/0072467509- introduction-to-computing - systems-from- bits - gates-to-c - beyond .pdf. clipped from Google - 10/2020. "Introduction" does just what its title says: it starts with bits (those 1's and 0's) and works all the way up to high level programming languages like C, hitting everything from Boolean and transistor logic to processor construction to assembly language in between. The book even comes with simulation software for a "computer" that allows the reader to practice and experience low-level programming and interaction without having to work on their own real computers. Spend a few weeks getting to know C before diving into this excellent book. May even be worth getting to know a little bit of the C ecosystem so you can do things like look at the assembly language that your C program produces. Read more. Helpful. introduction to computing systems The McGraw-Hill Companies Mc Graw Hill INTRODUCTION TO COMPUTING SYSTEMS: FROM BITS AND GATES TO C AND BEYOND SECOND EDITION International Edition 2005 Exclusive rights by McGraw-Hill Education (Asia), for manufacture and export. This book cannot be re-exported from the country to which it is sold by McGraw-Hill. The International Edition is not available in North America. 1.5 T w o V e r y I m p o r t a n t Ideas 9 2.7 Other Representations 36 1.6 Computers as Universal Computational 2.7.1 The Bit Vector 36 Devices 9 2.7.2 Floating Point Data Type 37 1.7 How Do W e Get the Electrons t o Do the 2.7.3 A S. C I I Codes 40 Work?