CS208 – Computer Science Fundamentals
Student Module

NOTE: For first night assignment and required class materials, see course overview.

Course Description

CS 208. COMPUTER SCIENCE FUNDAMENTALS (3). Provides an introduction for Computer Science/Computer Information Systems majors/minors. Includes hardware and software systems, numbering systems, mathematical and logical binary operations, and basic concepts of computer organization. Introduces program development environments using Pascal.

Learning Topics

Learning Topic #1: Course Introduction
Learning Topic #2: Computer Hardware
Learning Topic #3: Number Systems
Learning Topic #4: Codes for Data Representation
Learning Topic #5: Operations on Binary Numbers
Learning Topic #6: Signed Numbers
Learning Topic #7: Floating Point Numbers
Learning Topic #8: System Software
Learning Topic #9: Programming
Learning Topic #10: An Introduction to Pascal
Learning Topic #11: Assembly Language Concepts
Learning Topic #12: IS Analysis and Design
Learning Topic #13: Files and Databases
Learning Topic #14: Communications/Networks
Learning Topic #15: Ethics, Privacy, and Security
Learning Topic #1: Course Introduction

This topic is used to introduce students to the philosophy and contents of *Computer Science Fundamentals*, using this learning module as a reference. The scope and rationale for the course will be discussed in detail.

**Topic Outcomes**

Upon completion of this topic the student will be able to:

- Define the overall purpose of the course.
- Reference the learning module to identify course content.
- Access additional resources for further study of various topics.
- Understand the administrative aspects of the course.
- Understand the manner in which course assignments are made and evaluated.
- Identify the grading criteria used to evaluate student course work.
- Develop criteria as to capability/desire to pursue further studies in the Computer Science/Computer Information Systems disciplines.

**Self-Assessment Questions**

- What is meant by “computer literacy”?  
- Are grade- and high-schoolers more computer literate than their prior generation?  
- What is the difference between Computer Science and Computer Information Systems?  
- How can an individual determine his/her potential for success in a computer-related career?
Learning Topic #2: Computer Hardware

This topic includes information concerning the basic physical components of computers, how these components are integrated to facilitate processing, and various coding schemes used both to provide instructions to computers and store data. Emphasis is on the microcomputer (personal computer).

Topic Outcomes

Upon completion of this topic the student will be able to:

- Identify the basic hardware components of microcomputer systems.
- Explain the use of coding schemes for data processing.
- Describe the functions of the CPU, RAM, and registers.
- Explain the impact of RAM capacity and processor speed on computing power.

Self-Assessment Questions

- Identify the major factors in the proliferation of the personal computer.
- Do mainframe computers have a future?
- What is your understanding of data processing “efficiency”?
- Which hardware components have the greatest impact on this efficiency?
- Which coding schemes do you find easiest/hardest to understand? Why?
Learning Topic #3: Number Systems

This topic includes information concerning the various number systems associated with computer processing and data storage. While the binary number system (base 2: 1’s and 0’s) is basic to the operation of the computer (machine language), the hexadecimal (base 16) and octal (base 8) systems will also be discussed. The hexadecimal, and less frequently the octal systems, are used to translate binary numbers to numbers more easily read by the programmer. Each of these systems will be discussed in terms of the decimal number system.

Topic Outcomes

Upon completion of this topic the student will be able to:

- Identify the key words and terms associated with computer related number systems.
- Convert numbers from one base to another (decimal to binary, binary to hexadecimal, decimal to hexadecimal, etc.).
- Explain why the binary number system is uniquely suited to represent the internal processing circuitry used in computers.
- Understand the uses of the hexadecimal and octal number systems.

Self-Assessment Questions

- Have you used number systems other than the decimal system in the past?
- There is now an emphasis on computers in the grade- and high-schools. Should students at those educational levels be taught the non-decimal number systems contained in this learning topic?
- Can you identify computer related jobs where there may be no reason to understand other than the decimal number system?
- Can you identify non-computer-related areas where one or more of the topic number systems can be used?
Learning Topic #4: Codes for Data Representation

This topic includes information concerning the representation of character data within the computer. As indicated in Learning Topic #3, each bit can have two states, on or off. These states can be used to represent data as follows: no may be represented by off (0); yes may be represented by on (1). As the number of bits is expanded, the number of states available for representing data is expanded. In general, whenever a bit is added, the number of available states is doubled. In addition to representing numbers, therefore, binary values can also be used to represent characters. The American Standard Code for Information Interchange (ASCII) is a character set used with most microcomputers. This character set will be emphasized.

Topic Outcomes

Upon completion of this topic the student will be able to:

- Define data representation.
- Differentiate between numeric and character data.
- Explain how codes are used to facilitate communication between human users and computers.
- Identify the more common coding systems.
- Encode and decode ASCII character strings.

Self-Assessment Questions

- What are some codes used in everyday experiences?
- Can you relate codes from your own experiences to those used in computers?
- Which of the more common coding systems do you find easiest to understand?
- What is a common use for “strings”?
Learning Topic #5: Operations on Binary Numbers

The base-10 decimal system which we use every day uses the ten digits, 0 through 9. Since the only digits available to represent the electrical states in a computer are 0 and 1, we will illustrate how these base-2 numbers are manipulated to allow the computer to do meaningful work. Just as the decimal system uses the position of a digit within a number to determine the value of the individual digit, so too, with a binary number. In this unit the arithmetic operations of addition, subtraction, multiplication, and division will be explored for binary numbers. Additionally, logical operators, such as AND and OR, and shift operators, which move data, will also be covered.

Topic Outcomes

Upon completion of this topic the student will be able to:

• Describe the positional values of binary numbers.
• Perform binary addition, subtraction, multiplication, and division.
• Explain the OR, XOR, AND, NOT, logical operators.
• Explain the SHL and SHR shift operators.

Self-Assessment Questions

• There are only two numbers in the binary number system. Do you find it easier or more difficult to perform arithmetic operations with this system than with the decimal number system?
• Discuss some real-world situations in which logical operations are employed.
Learning Topic #6: Signed Numbers

Previous units have covered unsigned numbers. There are a number of schemes for representing positive and negative numbers in binary format. Two of the more common in use are the sign-magnitude representation and the twos-complement representation. This unit discusses these representations as well as addition and subtraction for twos-complement numbers.

Topic Outcomes

Upon completion of this topic the student will be able to:

- Identify the sign-magnitude representation of a binary number and the placement of its sign.
- Convert sign-magnitude numbers to ones-complement numbers.
- Convert ones-complement numbers to twos-complement numbers.
- Perform arithmetic and sign-change operations with twos-complement numbers.

Self-Assessment Questions

- Differentiate between the sign-magnitude representation and the twos-complement representation of binary numbers.
- What are the drawbacks of the ones- and twos-complement representations?
- What are the major advantages of the twos-complement representation?
Learning Topic #7: Floating Point Numbers

In the decimal system, a decimal point separates the whole numbers from the fractional part. Other systems also use a point to separate these parts. Henceforth, this point will be known as a \textit{radix point}, since its use is not limited to the decimal numbering system. In the decimal system, numbers are often written in \textit{scientific} or \textit{floating point} notation. The significant digits of a number are always written with the radix point between the first and second digits; this part of the number is called the \textit{significand}. The significand is multiplied by a power of 10 to move the radix point to the desired position. Binary numbers can also be written in floating point format in much the same manner as decimal numbers, except that powers of two, rather than powers of 10 are used to shift the radix point. This unit will cover both decimal and binary floating point numbers, their conversion from one to the other, and the method for storing binary floating point numbers in computer memory.

Topic Outcomes

Upon completion of this topic the student will be able to:

- Convert decimal numbers to their floating point format.
- Convert decimal floating point numbers to binary floating point numbers.
- Convert binary floating point numbers to decimal numbers.
- Convert binary floating point numbers to a format suitable for machine storage.

Self-Assessment Questions

- What is the need for floating point representation of numbers?
- Discuss the differences in the size of numbers available between the sign-magnitude and floating point representations of numbers.
- Discuss the advantages of \textit{biased exponents}. 
Learning Topic #8: System Software

This topic introduces the software which provides the basic operating functions of the computer. This underlying software allows the running of applications programs (word processors, spreadsheets, databases, etc.) and also provides the capability for users to interact with the computer. Included among the system software are the following categories: BIOS (basic input/output system), operating systems, data managers, communication protocols, drivers, utility programs, and language translators. While this list is by no means exhaustive, the emphasis of this unit is on operating systems.

Topic Outcomes

Upon completion of this topic the student will be able to:

- Define the various components of system software.
- Describe the basic system software interfaces.
- Describe the more common operating systems.

Self-Assessment Questions

- Discuss the term platform.
- Why do we have more than one computer operating system?
- Rank the various system functions in order of importance.
- Contrast multi-processing, time-sharing, and multi-processing.
- Contrast the various Microsoft operating systems.
Learning Topic #9: Programming

In simple terms, a program is a series of instructions which direct a computer to carry out a series of operations. The instructions consist of a series of statements written in a programming language such as BASIC. Programming, on the other hand, involves much more than writing instructions. Programming is, above all else, one method of problem solving. It uses a series of steps, called algorithms, to solve a problem. There are five well-defined steps in the programming process: identify the problem; design the solution (program); code the program using a programming language; test the program; document and maintain the program. As we will see in a later learning topic, programming is part of a more comprehensive methodology called the systems development life cycle (SDLC).

Topic Outcomes

Upon completion of this topic the student will be able to:

- Explain what a program is.
- Explain the five steps in traditional programming.
- Discuss the concepts of structured programming design.
- Identify the five generations of computer languages.
- Differentiate among assembler, compiler, and interpreter.
- Identify some traditional programming languages.
- Briefly describe: object-oriented and visual programming; HTML; VRML; Java.

Self-Assessment Questions

- How do third-generation languages differ from first- and second- generation languages?
- What was the reason for the development of higher-level languages?
- Explain the importance of design and program documentation.
- Are visual programs easier to create than traditional programs?
Learning Topic #10: An Introduction to Pascal

Pascal is a high-level computer language named after the 17th century French philosopher and mathematician, Blaise Pascal. Its author, Niklaus Wirth, a professor of computer science at the Swiss Federal Technical University at the time, began writing the language in 1968 as a tool for teaching his computer science students sound programming practices. This topic will introduce the most basic of Pascal’s language elements with an emphasis on the application of problem solving techniques.

Topic Outcomes

Upon completion of this topic the student will be able to:

- Describe the three components of a Pascal program.
- Differentiate between program variables and constants.
- Demonstrate basic looping techniques.
- Describe the use of program modules (procedures).
- Perform basic text file processing operations.
- Explain how this language is used to implement structured programming methods.
- Write several moderately short Pascal programs.
- Implement a structured program.

Self-Assessment Questions

- Why is Pascal more suited for teaching the theory and techniques of programming, but not for writing practical applications?
- What, if anything, do you know of the current emphasis to use C/C++ as the first programming language in a computer science/computer information systems curriculum?
- Do you find it easy or difficult to implement structured programs in Pascal using the structured design methodology?
Learning Topic #11: Assembly Language Concepts

There are a wide variety of computer languages which allow users to tailor programs to meet specific needs. Some are quite easy to work with, using English-like instructions, while others are quite cryptic. The former are often referred to as higher level languages (BASIC, COBOL, Pascal). This topic explores a low-level language known as assembly language, which may be considered one step above the ones and zeros of the machine language with which a computer actually works. To further the understanding of assembly language, a small instruction set will be specified and used to write code. This code will allow a number of simple operations similar in nature to those performed by real assembly language instruction sets.

Topic Outcomes

Upon completion of this topic the student will be able to:

• Explain the relationship between machine code and assembly language code.
• Explain how the use of mnemonics facilitates the coding process.
• Explain how the compiler distinguishes between data and instructions.
• Describe the operation of several assembly language instructions.
• Write short assembly language programs using the specified instruction set.

Self-Assessment Questions

• Discuss the advantages and disadvantages of programming at the assembly language level.
• Why should one become familiar with assembly language programming if programming only with higher level languages?
• Do you find it easier to relate non-decimal number systems to assembly language or to higher order level languages?
• Can the basic concepts of problem solving and structured programming be applied using assembly languages?
Learning Topic #12: IS Analysis and Design

What is a system? The dictionary provides several definitions. For example, Webster’s Collegiate Dictionary, 10th Edition, includes the following: “a regularly interacting or interdependent group of items forming a unified whole”; “a group of devices or artificial objects or an organization forming a network esp. for distributing something or serving a common purpose”. It further defines information science: “the collection, classification, storage, retrieval, and dissemination of recorded knowledge treated as both a pure and applied science”. This topic will introduce information systems in a manner which incorporates the concepts in these definitions, and which provides a step-by-step approach to developing and maintaining such systems using a methodology known as the systems development life cycle (SDLC).

Topic Outcomes

Upon completion of this topic the student will be able to:

- Describe the role of the user in the systems development life cycle.
- Describe the six phases of the systems development life cycle.
- List some techniques for gathering and analyzing data describing the current system.
- Describe some software tools used by systems analysts.
- Describe one or more basic approaches to implementing a new computer-based information system.

Self-Assessment Questions

- Why do some systems fail?
- How good are case tools?
- What is the purpose of joint applications development (JAD)?
- What is the difference between a structured and non-structured interview?
Learning Topic #13: Files and Databases

In general, files are collections of data records stored on tape or disk. When stored on tape, records can be accessed only in a sequential mode. When stored on disk, these records may be accessed either sequentially or directly (random access). File management systems (FMS) are collections of programs for manipulating single files. Files management systems were prevalent in the earlier days of business computing. A database, on the other hand, may be thought of as a collection of related files, usually stored on disk, which may be manipulated, in an integrated manner, by a database management system (DBMS). The emphasis for this topic will be on the nature of various database management systems.

Topic Outcomes

Upon completion of this topic the student will be able to:

- Describe the parts of the data hierarchy.
- Distinguish between batch and real-time processing.
- Distinguish between online and offline data storage.
- Explain sequential, direct, and indexed sequential access storage.
- Differentiate between file management systems and database management systems.

Self-Assessment Questions

- What are the advantages of online processing over batch processing.
- What are the advantages of batch processing over online processing?
- What are the main advantages of traditional file management systems?
- What is a query language?
**Learning Topic #14: Communications/Networks**

This topic will introduce communications technology, the means by which data is transmitted, electronically, from one device to another. The discussion will include analog and digital signals, modems, channels, and communications software. Additionally, computer networks, such as local area networks (LANs) and wide area networks (WANs) will be introduced.

**Topic Outcomes**

Upon completion of this topic the student will be able to:

- Explain basic communications principles.
- Explain various communications channels.
- Discuss factors affecting communications among devices.
- Identify the three main types of networks.
- Identify the five basic types of local area networks.

**Self-Assessment Questions**

- What impact does **modem** speed have upon **Internet** users?
- What are some of the benefits of **microwave** and **satellite** communications?
- What are some uses for the **Global Positioning System**?
- Discuss the relationship between transmission rates and cost.
- Discuss the levels of the **ISO** standard.
- What are some advantages of **networks**?
- What is the significance of the **electromagnetic spectrum**?
Learning Topic #15: Ethics, Privacy, and Security

This topic discusses the important subjects of ethics, privacy, and security in an age of pervasive technology. Webster’s Collegiate Dictionary, 10th Edition, includes the following definitions of ethic: “the discipline dealing with what is good and bad and with moral duty and obligation”; “the principles of conduct governing an individual or a group (professional ethics)”. The proliferation of technology, particularly computer systems, raises new kinds of ethical problems. Just as concepts of ethics are impacted by the age of technology, so to are those of privacy. Federal and state laws provide some protection to individuals and corporations against the “invasion” of privacy, but misuse of electronically stored data cannot be totally prevented. Related to ethics and privacy considerations is the subject of security. Approaches to safeguarding computer and communication systems will also be introduced under this topic.

Topic Outcomes

Upon completion of this topic the student will be able to:

- Describe the basic concepts of ethics, privacy, and security as they relate to computers.
- Describe the intellectual property matters relating to computer software and networks.
- Explain how computers can be used to alter computer stored or generated materials.
- Describe the major threats to computer and communications systems.
- Explain how computer and communications systems can be made more secure.
- Understand some of the social issues related to computing.

Self-Assessment Questions

- Discuss new ethical problems arising from the age of technology.
- Discuss how computers and networks can be used to invade privacy.
- How is free speech viewed relative to networks such as the Internet?
- Discuss some computer/network related social issues.
RESOURCES

Bibliography

• Regis University, CS208 Computer Science Fundamentals Supplemental Materials.

Internet

• For America On Line (AOL) subscribers: The Computing Channel offers both fee and free computing courses. Select Online Classroom and browse from there.
• Amazon.com at http://www.amazon.com offers a wide variety of computing books. Use the following to find some of these titles: COMPUTER SCIENCE; COMPUTER SCIENCE FUNDAMENTALS; INFORMATION SYSTEMS; COMPUTER INFORMATION SYSTEMS.

Other

• Small assembly language compiler provided by Regis University. Compliments of Mark Sanders, Co-lead for CS/CIS.