Georgia Perimeter College  
Mathematics/Engineering/Computer Science Division  

TEACHING GUIDE  
CSCI 1300 – Introduction to Computer Science

I. Course Title: Introduction to Computer Science

II. Prerequisite: Exit: Learning Support, ESL, and completion of MATH 1111 (College Algebra) with a C or better.

II. Text:  
Invitation to Computer Science, 7th Edition (w/CourseMate/eBkLabMan)  
G. Michael Schneider | Judith Gersting  
ISBN: 9781305720602  
Publisher: Cengage Learning

IV. Catalog Description: This course provides an overview of selected major areas of current computing technology, organization and use. Topics surveyed include the history of computing, data representation and storage, hardware and software organization, communications technologies, ethical and social issues, and fundamental problem-solving and programming skills. Hands-on projects enhance and reinforce the ideas presented in class. This course is intended for computer science majors, as well as mathematics and science-based majors. Students may NOT receive credit for both CSCI 1100 and CSCI 1300.

V. Course Objective: The course provides students with knowledge of computer science concepts and computing skills through extensive problem-solving and project opportunities.

VI. General Notes: The instructor may decide on the sequence of topics covered, but must cover appropriate sections in the text to comply with the Common Course Outline. Assignments are left to the discretion of the instructor. However, use of the eBook lab manual that accompanies this text is highly encouraged; the instructor may selectively pick and choose among the laboratory exercises to be offered.

VII. Course Content:

TOPICS TO BE COVERED:  

History and Vocabulary of Computers  
- What is a Computer?  
- Types of Computers  
- Common terms

SUGGESTED CHAPTERS:

1
• Historical issues

**Problem-Solving, Algorithms, and Algorithm Efficiency**

2.1-2.3.3, 3.1-3.4.2

• Pseudocode
• Control Structures
• Algorithm Development & Examples
• Algorithmic Problem Solving:
  o Sequential Search
  o Find largest Algorithm
• Algorithm Efficiency and Order of Magnitude

**Data Representation and Storage**

4.1-4.2

• Number systems
• Binary
• Octal
• Hexadecimal
• Text
  o ASCII
  o Unicode
• Sound
• Graphics
  o Raster
  o Vector
• Video
• Data Compression
  o Run-Length Encoding
  o Variable-Length Encoding (Huffman)
• Transistors

**Computer Hardware Concepts**

4.3-4.6, 5.1-5.5

• Boolean Logic and Gates
• Circuits and Circuit Construction
• Control Circuits (optional)
• Von Neumann Architecture
• Memory
  o ROM
  o RAM
  o Cache
• Input/Output and Mass Storage
  o Input devices
  o Output devices
  o External Storage technologies
• Arithmetic-Logic Unit
  o Registers
  o Circuitry
- Control Unit
  - Program Counter
  - Instruction Register
  - Fetch/Decode/Execute Cycle
  - Machine Language Instructions
- Parallel Architectures

**Systems Software & Operating Systems** 6.1-6.2, 6.3(optional), 6.4
- Functions
- Types
- Historical Overview

**Computer Networks, the Internet, and the World Wide Web** 7.1-7.6
- Basic Networking Concepts
- Computer Network types
- Network topologies
- The Internet
- The World Wide Web

**Information Security** 8.1-8.7
- Threats & Defenses
- Authentication and Authorization
- Threats from the Network
- Encryption & Encryption Algorithms

**Programming Concepts & Problem-Solving** 9.1-9.7
- Imperative vs. Object-Oriented Languages
- Program Structure
- Data Types and Declarations
- Input and output
- Assignments
- Computation
- Selection and Repetition
- Modular programming (optional)

**Applications Software** 14.1-14.2
- E-Commerce
- Databases 14.3
- Spreadsheets Not supported by text (opt.)

**Social and Ethical Issues** 17.1-17.5
- Privacy and information access
- Censorship
- Intellectual property rights
• Computer crime
• Professional responsibilities
• On-line communities
• Technology in everyday life

III. Evaluation Methods

Details of grade determination are left to the instructor with the approval of the Department Head. Exams, assignments, and a final exam prepared by individual instructors will be used to determine the course grade. The course grade must weigh examinations for at least 50% of the grade and laboratory and programming assignments for not more than 50% of the grade. At least eight to ten project-based assignments must be assigned. Testing must consist of at least two one-hour examinations and a comprehensive final examination. The final examination must be weighted at not less than 25% nor more than 35%.

Effective Date: Fall 2015

Approved Date: Fall 2015