

**FORM VERSION: 5/95**  
**DATE REVISED: 3/11/2000**

**PALOMAR COLLEGE**  
**COURSE OUTLINE OF RECORD FOR**  
**DEGREE CREDIT COURSE**

  x   Transfer course      x   A.A. degree applicable course

**COURSE NUMBER AND TITLE:**            **CSIS 160 Survey of Computer Science**

**UNIT VALUE:** 4

**MINIMUM NUMBER OF SEMESTER HOURS:** 80

**BASIC SKILLS REQUIREMENTS:**        Appropriate language and computational skills

**ENTRANCE REQUIREMENTS:**

**PREREQUISITE:**        None

**COREQUISITE:**        None

**RECOMMENDED PREPARATION:** None

**SCOPE OF COURSE:**

An overview of the discipline of computer science including such topics as the history of computer science; machine architecture; data storage and manipulation; operating software engineering; data structures; database and information retrieval; data communications; artificial intelligence; theory of computation; social legal and ethical issues. Includes hands-on laboratory experience reinforcing the lecture material.

**SPECIFIC COURSE OBJECTIVES:** Successful students will be able to:

1. Specify what an algorithm is; be able to trace through a given (simple) algorithm, give examples of algorithms, and find flaws in poorly-written algorithms.
2. Explain the science that lies behind the computer applications with which most students are familiar
3. For simple to moderate algorithms the student will translate in both directions between a natural language description of the algorithm and pseudo-code for the algorithm.
4. List and explain the attributes of an algorithm.
5. Explain the reason binary numbers are used.
6. Translate between decimal, hexadecimal, and binary and do simple hexadecimal and binary calculations.
7. Determine the truth value of Boolean expressions.
8. List and describe the subsystems of a modern digital computer.
9. Explain by example the difference between RISC and CISC machines.
10. Describe the stored-program concept.
11. Draw and explain the Fetch-Decode-Execute cycle.
12. Specify the advances made in operating systems with each computer generation.
13. Explain the way an operating system manages processor time and solves problems like deadlock.
14. Identify the importance of computer security and understand and be able to distinguish between the use of passwords and authorization lists.
15. Identify the different roles and functions of an operating systems.
16. Demonstrate by example with pseudo-code the purpose and semantics of sequential, iterative, and conditional control constructs.
17. Represent with pseudo-code a recursive algorithm.
18. Distinguish the appropriate application of assembly and high-level programming languages to the solution of a problem.
19. Identify and demonstrate an understanding of the need for different language design philosophies.
20. Delineate the differences between the different models of programming language: imperative, functional, logic, object orientated, and parallel.
21. Describe the stages of the Software Life Cycle.
22. Elucidate the components of Modularity: Coupling, Side Effects, and Cohesion.
23. Illustrate the following software development tools and techniques: Top-Down Design, Bottom-Up Design, Dataflow Diagrams, Entity-Relationship Diagrams, and Data Dictionaries.
24. Explain by example ,utilizing pseudo-code, the following data structures: Arrays, Lists, Stacks, Queues, Trees, Abstract Data Types, and their representation in Object-Orientated Programming environment.
25. Demonstrate an understanding of sequential, text, hashed, and indexed files.
26. Explain the relationship between an operating system and the file structures supported by that system.
27. Explain general issues of Database Structure, the Layered Approach to Database Implementation, the Relational Model, Hierarchical Model, Network Model, Object-Orientated Model, and issues of Concurrency Control.
28. Describe some of the Philosophical issues associated with Artificial Intelligence such as

Machine vs. Humans, Performance vs. Simulation, Intelligence as an Interior Characteristic, and what is an AI Intelligent Machine.

29. Elucidate the use of Heuristics, Control System Activities, Reasoning, Image Analysis, and Artificial Neural Networks in the field of Artificial Intelligence.
30. Demonstrate an understanding of the basic ideas associated with the Theory of Computation such as Turing Machines, Computable Functions, Non-computable Functions, Complexity and its Measure, and Problem Classification.

## **CONTENT IN TERMS OF SPECIFIC BODY OF KNOWLEDGE:**

### **I. Introduction to Computer Science**

- A. History of Computer Science

### **II. Machine Architecture / Hardware**

- A. Data Representation
- B. Data Storage
- C. Data Manipulation
- D. Number Systems
  1. Decimal
  2. Binary
  3. Octal
  4. Hexadecimal
- E. Binary Logic
- F. Circuits and Gates
- G. Computer Systems Architecture
- H. Von Neumann Architecture
- I. CISC / RISC
- J. Parallel Processing
- K. Machine and Assembler Language

### **III. Operating Systems**

- A. History, Evolution, and Philosophy
- B. Functions of the Operating System
- C. Time Sharing
- D. Resource Allocation
- E. User Interfaces
- F. Tasks and Processes
- G. Scheduling
- H. Synchronization
- I. Deadlock
- J. Physical and Virtual Memory
- K. File Systems
- L. Security and Protection

### **IV. Algorithms**

- A. Algorithms
- B. Pseudo Code
- C. Sequence, Selection, Iteration

- D. Problem Solving Methodologies
- E. Greedy Algorithms
- F. Divide and Conquer
- G. Backtracking Algorithms
- H. Algorithmic Efficiency and Correctness

**V. High-Level Programming Languages**

- A. History and Overview
- B. Procedural Programming
- C. Functional Programming
- E. Logic Programming
- F. Object-Oriented Programming
- G. Parallel Programming

**VI. Compilers and Language Translators**

- A. Interpreters vs Compilers
- B. Grammars and Languages
- C. Finite State Automata
- D. Regular Expressions
- E. The Compilation Process
- F. Lexical Analysis
- G. Parsing
- H. Code Generation
- I. Optimization

**VII. Software Engineering**

- A. Software Development Process
- B. Software Life Cycle Models
- C. Development Tools and Techniques
- D. Documentation
- E. Software Reliability Issues

**VIII. Data Structures**

- A. Abstract Data Types
- B. Basic Data Structures
  - 1. Arrays
  - 2. Stacks
  - 3. Queues
  - 4. Lists
  - 5. Trees
  - 6. Space vs Time Tradeoffs
  - 7. Searching
  - 8. Sorting
- C. Memory Management
  - 1. Garbage Collection

**IX. Database and Information Retrieval**

- A. File Management
- B. Sequential Files
- C. Text Files
- D. Indexed Files
- E. Hashed Files
- F. Data Organization
- G. Database Management Systems
- H. Relational Model
- I. Network Model
- J. Hierarchical Model
- K. Object-Oriented Model

**X. Data Communication**

- A. Data Communications
- B. ISO/OSI and Internet Models
- C. Data Transmission
- D. Transmission Media
- E. Data Encoding
- F. Wide-Area Networks Technologies
- G. LAN and Intranet Technologies
- H. Internet Information Services
- I. Distributed Systems and Their Applications
- J. Network Security Issues

**XI. Artificial Intelligence**

- A. History and Application
- B. Knowledge Representation
- C. Game Trees
- D. Heuristics
- E. Expert Systems
- F. Neural Networks
- G. Decision Support Systems
- H. Robotics

**XII. Theory of Computation**

- A. Turing Machines
- B. Church-Turing Thesis
- C. Computability
- D. N-P Complete Problems

**XIII. Social, Legal and Ethical Issues**

- A. Historical and Social Context of Computing
- B. Computer Crime
- C. Responsibilities of the Computing Professional
- D. Risks and Liabilities
- E. Intellectual Property

**REQUIRED READING:**

Brookshear, Glen. Computer Science An Overview, 5th edition. Redwood City: Benjamin Cummings Publishing Co., 1996.

**SUGGESTED READING:** None

**REQUIRED WRITING:**

Problem solving exercises are assigned, requiring students to complete five or six lab exercises. Each lab will consist of a hands-on exercise applying theory principles learned in class. Labs must be well documented (at least one paragraph) in terms of their overall design goals.

**OUTSIDE ASSIGNMENTS:**

**Students are expected to spend a minimum of three hours per unit per week in class and on outside assignments, prorated for short term classes.**

Outside assignments will include completion of lab work, assigned readings, and homework problems.

**INSTRUCTIONAL METHODOLOGY:**

**Check all that apply:**

- lecture
- laboratory
- lecture-laboratory combination
- directed study

**This course may be offered as a distance education course and meets Title 5 regulations 55370, 55372, 55374, 55376, 55378, and 55380. Yes  No**

**If yes, check all that apply:**

- telecourse
- mediated instruction
- computer assisted instruction

**GRADING POLICY AND STANDARDS** (include methods of determining whether the stated objectives have been met by students):

Grades for courses are based upon final examinations, mid-term examinations, other tests, assignments, projects, and participation. Faculty will inform students of their grading policy at the beginning of each semester.

SIGNATURES ON FILE
--------------------