L. Mayfield
FA 2023

COMP235 Course Competencies

Knowledge Areas

1. SDF / Fundamental Programming Concepts & Practices [in C]
    a. Variables, primitive types, expression evaluation, etc.
    b. Pointers
    c. Conditionals and Loops
    d. Functions and Procedures, Pass-by-Value and Pass-By-Reference, variable scope
    e. Arrays and C-Strings
    f. Errors, testing, and debugging
    g. Reading and Understanding Code
2. AR / Machine-Level Data Representation
    a. Binary, Octal, and Hexadecimal Numbers
    b. Unsigned and Two's Complement Integers
    c. Integer Arithmetic and Associated Errors
    d. Floating-Point
    e. Floating-Point Arithmetic and Associated Errors
3. AR / Assembly-Level Machine Organization
    a. Accessing Information
    b. Arithmetic and Logic Operations
    c. Control
    d. Procedures and the Run-Time Stack
    e. Array Allocation and Access
    f. Pointers
4. AR / Memory Hierarchy
    a. Locality
    b. Caching Techniques
    c. Virtual Memory
5. OS / Memory Management
    a. Memory Mapping
    b. Dynamic Memory Allocation
    c. Garbage Collection
6. SF / Basic Systems Concepts and Overview
7. SF / Resource Allocation and Scheduling
8. SF / System Performance
    a. Latency
    b. Locality Effects and Caches
    c. Cache Coherence
    d. Virtual Memory Implementation
9. SF / Performance Evaluation

Course competencies derived from and otherwise based on ACM curricular guidelines. By and large, these course competencies borrow directly from the March 2023 Version Beta draft of the *Computer Science Curricula 2023*. For more information see https://csed.acm.org/cs2023-beta/

    a. Amdahl's Law
    b. Big-O Analysis
10. SF / System Reliability
    a. Bugs and Faults
    b. Redundancy and Reliability
11. SF / System Security
12. SF / System Design

Professional Dispositions

- **Meticulousness**: Pay attention to details and different perspectives when learning about and evaluating systems.
- **Adaptiveness:** Be flexible and adaptive when designing systems. Different systems have different requirements, constraints, and working scenarios. Be able to make appropriate design decisions correspondingly.
- **Self-Directed:** Become self-motivated to acquire complementary knowledge from system documentation.
- **Proactive:** Be proactive and independent in order to navigate and integrate knowledge from different knowledge areas to understand the underlying computer system.
- **Inventive:** Look beyond simple solutions to computer architecture and system design issues and leverage architecture-specific features whenever possible.

Skill's Hierarchy

1. **Explain** - define, describe, discuss, enumerate, express, identify, indicate, list, name, select, state, summarize, tabulate, translate
2. **Apply -** backup, calculate, compute, configure, debug, experiment, install, iterate, interpret, manipulate, map, measure, predict, randomize, restore, schedule, solve, test, trace
3. **Evaluate** - analyze, compare, classify, contrast, distinguish, categorize, differentiate, discriminate, order, prioritize, criticize, support, assess, choose, defend, rank
4. **Develop** - combine, compile, compose, create, design, generalize, integrate, modify, organize, produce, rewrite, refactor, write

Course competencies derived from and otherwise based on ACM curricular guidelines. By and large, these course competencies borrow directly from the March 2023 Version Beta draft of the *Computer Science Curricula 2023*. For more information see https://csed.acm.org/cs2023-beta/

Competencies

1. **Task: Assess the performance implications of cache memories in your application.**
    a. *Competency:* Critically analyze the performance of an application concerning caching issues and produce a report summarizing key results.
    b. *Required Knowledge:* AR/Memory Hierarchy, AR/Performance and Energy Efficiency, OS/Memory Management.
    c. *Required Skill:* Evaluate
2. **Task: Work on a program explicitly exploiting the underlying CPU cores and memory management system for improved performance.**
    a. *Competency:* Understand the computer architecture principles, parallelism, and how memory is managed for efficient data sharing.
    b. *Required Knowledge*: AR/Assembly Level Machine Organization, AR/Memory Hierarchy
    c. *Required Skill: Evaluate/Develop*
3. **Task: Document a system's design choices and proposed system hardware and software architecture.**
    a. *Competency:* Gather application requirements and propose a system architecture solution. Write effective technical documentation.
    b. *Required Knowledge:* AR/Machine-Level Data Representation, AR/Performance and Energy Efficiency, SEP/Professional Communication, SE/Product Requirements
    c. *Required Skill*: Explain/Evaluate.
4. **Task: Evaluate the performance of a given system**
    a. *Competency:* Given a system, evaluate key performance metrics (e.g. correctness, throughput, and average/tail latency) analytically and experimentally.
    b. *Required Knowledge:* SF/System Performance, SF/Performance Evaluation
    c. *Required Skill*: Apply/Evaluate
5. **Task: Find the performance bottleneck of a given system.**
    a. *Competency:* GIven a system and its target deployment environment, find it's performance bottleneck (e.g. memory, CPU, networking) through analytical derivation or experimental study.
    b. *Required Knowledge:* SF/System Performance, SF/Performance Evaluation, SF/System Design, SF/Overview of Computer Systems
    c. *Required Skill:* Apply/Evaluation/Develop
6. **Task: Find and fix bugs in a system.**
    a. *Competency:* Given a system, its deployed environment and its buggy symptoms, find bugs by using tools for program tracing, single stepping, and debugging, and fix them.
    b. *Required Knowledge:* SF/System Design, SF/System Reliability, SF/Performance Evaluation
    c. *Required Skill:* Apply/Evaluate/Develop

Course competencies derived from and otherwise based on ACM curricular guidelines. By and large, these course competencies borrow directly from the March 2023 Version Beta draft of the *Computer Science Curricula 2023*. For more information see https://csed.acm.org/cs2023-beta/