**CMPS 2020 Programming II: Data Structures and Algorithms**

**Catalog Description**

**CMPS 2020 Programming II: Data Structures and Algorithms (4)**

Builds on the foundation provided by CMPS 2010 to introduce the fundamental concepts of data structures and algorithms that proceed from them within the framework of object-oriented programming technology. Topics include: recursion, fundamental data structures (including lists, stacks, queues, hash tables, trees and graphs) and basics of algorithmic analysis. Necessary components of object-oriented programming method will be introduced. Each week lecture meets for 150 minutes and lab meets for 150 minutes. Prerequisite: CMPS 2010 with C- or higher.

**Prerequisites by Topic**

Fundamentals of any programming language including selective and repetitive constructors, concepts of subprograms (functions), and user-defined types.

**Units and Contact Time**

4 semester units. 3 units lecture (150 minutes), 1 unit lab (150 minutes).

**Type**

Required for CS, CE

**Required Textbook**

ADTs, Data Structures, and Problem Solving with C++, Second Edition, Larry Nyhoff. ISBN 0-13-14909-3

**Recommended Textbook and Other Supplemental Materials**

None

**Coordinator(s)**

Huaqing Wang

**Student Learning Outcomes**

ACM/IEEE Body of Knowledge Topics:

[AL1] *Object-oriented design*: problems are analyzed via Object-Based Design paradigms

 [AL2] *Basic algorithm design*: Searching, Sorting algorithms are studied in depth

 [AL1,2] *Algorithms and problem-solving*: Classic techniques for algorithm design; problem-solving in the

object-oriented paradigm; application of algorithm design techniques to a medium-sized project

[AL3] *Basic algorithmic analysis*: Asymptotic analysis of upper and average complexity bounds; identifying

 differences among best, average, and worst case behaviors; big "O" notation; standard complexity classes; empirical measurements of performance; time and space tradeoffs in algorithms

 [PF4] *Recursion*: The concept of recursion; recursive mathematical functions; simple recursive procedures,

 divide-and-conquer strategies; recursive backtracking; implementation of recursion; recursion on trees,

 and graphs

 [AL3] *Fundamental computing algorithms*: Searching algorithms, Sorting algorithms in contiguous array, link

and binary tree search and tree sorting.

 [PF3]*Fundamental data structures*: Pointers and references; linked list structures; implementation strategies

for stacks, queues, and hash table implementation strategies for trees.

**ABET Outcome Coverage**

3a. An ability to apply knowledge of computing and mathematics appropriate to the discipline.

3b. An ability to analyze a problem, and identify and deﬁne the computing requirements and

 speciﬁcations appropriate to its solution.

3c. An ability to design, implement and evaluate a computer-based system, process, component,

 or program to meet desired needs.

3j. An ability to apply mathematical foundations, algorithmic principles, and computer science

 theory in the modeling and design of computer-based systems in a way that demonstrates

 comprehension of the tradeoﬀs involved in design choices.

**Lecture Topics and Rough Schedule**

Week 1 Chapter 1. Software Engineering

Week 2 Chapter 2. Introduction to Abstract Data Types

Week 3 Chapter 3. Data Structures and Abstract Data Types

Week 4 Chapter 4. More about OOP and ADTs -- Classes

Week 5 Chapter 5. Standard Input/Output and String Classes.

Week 6 Chapter 6. Lists

 Week 7 Chapter 7. Stacks

Week 8  Chapter 8. Queues

Week 9 Chapter 9. ADT Implementations: Templates and Standard Containers. – Generic Classes

Week 10 Chapter 10. ADT Implementations: Recursion, Algorithm Analysis and Standard Algorithms

Week 11 Chapter 11. More Linking Up with Linked Lists

Week 12 Chapter 12. Searching: Binary Trees and Hash Tables

Week 13 Chapter 13. Sorting

Week 14 Chapter 14. OOP and ADTs

Week 15 Chapter 15. Trees

Week 16 16. Graphs and Digraphs

**Design Content Description**

Not applicable to this course.

**Prepared By**

Huaqing Wang on [date]

**Approval**

Approved by CEE/CS Department on [date]
Effective [term]