**CMPS 221 Programming Fundamentals**

**Catalog Description**

**CMPS 221 Programming Fundamentals (5)**

Introduces the fundamentals of procedural programming. Topics include: data types, control structures, functions, arrays, and standard and file I/O. The mechanics of compiling, linking, running, debugging and testing within a particular programming environment are covered. Ethical issues and a historical perspective of programming within the context of computer science as a discipline are given. Each week lecture meets for 200 minutes and lab meets for 150 minutes. Prerequisite: (1) MATH 85; or (2) other satisfaction of the Entry Level Mathematics requirement.

**Prerequisites by Topic**

Passing score on ELM OR satisfaction of the ELM exemptions AND a passing

score on the Pre-Calculus Readiness Test (see Class Schedule for details)

OR completion of math remediation

**Units and Contact Time**

5 quarter units. 4 units lecture (200 minutes), 1 unit lab (150 minutes).

**Type**

Required for CS

**Required Textbook**

6th edition: Starting Out with C++: From Control Structures through Objects,

Tony Gaddis. Publisher: Addison Wesley, 2009. ISBN: 0321545885, or 7th edition: Starting Out with C++: From Control Structures through Objects,

Tony Gaddis. Publisher: Addison Wesley, 2011. ISBN: 0132576252

**Recommended Textbook and Other Supplemental Materials**

None

**Coordinator(s)**

Huaqing Wang

**Student Learning Outcomes**

ACM/IEEE Body of Knowledge Topics:

(CS-PF1/CE-PRF1,2) Fundamental programming constructs and paradigms

(CS-PF2/CE-PRF3) Algorithms and problem solving

(CS-PF3/CE-PRF4) Data structures (Introduction: basic types, strings and arrays)

(CS-SP1/CE-PRF0) History of computing

**ABET Outcome Coverage**

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

requirements and specifications appropriate to its solution.

Laboratory and homework assignments will require analysis of the

problem for successful completion of the assignments.

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

process, component, or program to meet desired needs. An ability to understand

the analysis, design, and implementation of a computerized solution to a

real-life problem.

Laboratory and homework assignments require analyzing the presented

problems, designing a solution to those problems, and implementing the

solution in a high-level programming language.

**Lecture Topics and Rough Schedule**

Week 1 Chapter 1: Introduction to Computers and Programming Chapter 2: Introduction to C++ (2.1 - 2.8, 2.11,

2.13 - 2.14) Chapter 3: Expressions and Interactivity (3.1, 3.2, 3.4)

Week 2 Chapter 2: Introduction to C++ (2.9) Chapter 3: Expressions and Interactivity (3.2 - 3.5) Chapter 4:

Making Decisions (4.1 - 4.7, 4.9- 4.11) Chapter 5: Looping (5.1 - 5.3)

Week 3 Chapter 2: Introduction to C++ (2.12) Chapter 3: Expressions and Interactivity (3.6) Chapter 4: Making

Decisions (4.8, 4.12, 4.14, 4.15),Chapter 5: Looping (5.3 - 5.8, 5.11 - 5.13)

Week 4 Review - Chapters 1-5 & Sample Midterm Discussion. In class coding midterm (attendance required).

Week 5 Written midterm, Chapter 6: Functions (6.1 - 6.10, 6.14)

Week 6 Chapter 6: Functions (6.1 - 6.10, 6.14) continued

Week 7 Chapter 6: Functions (6.10 - 6.15), Chapter 10: Characters, Strings and the string Class (10.1 - 10.5)

Chapter 3: Formatted Input and File I/O (3.9, 3.14) Chapter 5: Using a Loop to Read Data from a File

(5.9)

Week 8 Chapter 7: Arrays (7.1 - 7.5, 7.7, 7.11)

Week 9 Chapter 9: Pointers (9.1 - 9.7, 9.9)

Week 10 Additional topics (e.g. Structured Data, Classes, Recursions)

**Design Content Description**

Not applicable to this course.

**Prepared By**

Huaqing Wang on [date]

**Approval**

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