**CMPS 2010 Programming I: Programming Fundamentals**

**Catalog Description**

**CMPS 2010 Programming I: Programming Fundamentals (5)**

Introduces the fundamentals of procedural programming and Object-Oriented Programming. Topics include: data types, control structures, functions, arrays, I/O, pointer and dynamic Memory allocation, features of object-Oriented Programming features. 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. **Prerequisite**: (1) MATH 0030; or (2) other satisfaction of the Entry Level Mathematics requirement.

**Prerequisites by Topic**

Passing score on ELM OR satisfaction of the ELM exemptions OR completion of math remediation

**Units and Contact Time**

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

**Type**

Required for CS, CE, EE

**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++

Week 2 Chapter 3 Expressions and Interactivity

Week 3 Chapter 4 Making Decisions

Week 4 Chapter 5 Loops and Files

Week 5 Chapter 6 Functions

Week 6 Chapter 7 Arrays

Week 7 Chapter 8 Searching and Sorting Arrays

Week 8 Chapter 9 Pointers

Week 9 Chapter 10 Characters, C-Strings, and More About the string Class

Week 10 Chapter 11 Structured Data

Week 11 Chapter 12 Advanced File Operations

Week 12 Chapter 13 Introduction to Classes

Week 13 Chapter 14 More About Classes

Week 14 Chapter 15 Inheritance, Polymorphism, and Virtual Functions

Week 15 Chapter 16 Templates and Chapter 19 Recursion

**Design Content Description**

Not applicable to this course.

**Prepared By**

Huaqing Wang on [date]

**Approval**

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