**CMPS 222 Object-Oriented Programming**

**Catalog Description**

**CMPS 222 Object-Oriented Programming (5)**

Builds on foundation provided by CMPS 221 to introduce the concepts of object-oriented programming. The course focuses on the definition and use of classes and the fundamentals of object-oriented design. Other topics include: an overview of programming language principles, basic searching and sorting techniques, and an introduction to software engineering issues. Each Week lecture meets for 200 minutes and lab meets for 150 minutes. Prerequisite: CMPS 221 with C- or higher.

**Prerequisites by Topic**

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

**Units and Contact Time**

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

**Type**

Required for CS

**Required Textbook**

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:

|  |  |
| --- | --- |
| (PL1) Overview of programming languages | .10 |
| (PF2) Algorithms and problem-solving | .20 |
| (PL6) Object-oriented programming | 3.5 |
| (PF3) Fundamental data structures | .25 |
| (AL3) Fundamental computing algorithms | .25 |
| (SE1) Software design | .25 |

**ABET Outcome Coverage**

The course is designed to introduce the Object-Oriented programming methodology which is used in most all new languages. The courses also combined top-down designed method to enhance student ability in designing and implementing software application abilities. Specially, the following ABET defined student outcomes are defined.

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.

**Lecture Topics and Rough Schedule**

Chapter 9 Pointers - review and enhancement

Chapter 10 Characters, Strings and the string Class - review

Chapter 11 Structured Data - review and enhancement

Chapter 12 Advanced File Operations

Chapter 13 Introduction to Classes –review and enhancement

Chapter 14 More About Classes

Chapter 15 Inheritance, Polymorphism, and Virtual Functions

Chapter 16 Exceptions, Templates, and the Standard Template Library

Chapter 17 Linked Lists

Chapter 18 Stacks and Queues

Chapter 19 Recursion

Week 1 Review and enhance concepts related to functions including function prototypes, forward declaration,

parameter-passing methods, return types, function overloading and default parameters. More on static

and dynamic arrays.

Week 2 Review and enhance concepts on pointer, data structures, data encapsulation and abstract data type, and

introduction of data encapsulation and abstract data types. Review and enhance Chapter 9 Pointers and

Chapter 11 Structured Data

Week 3 Chapter 13 Introduction to Classes –review and enhancement Review 13.1 – 13.5, Lecture on 13.6 – 13.2

The concepts on constructors, destructor, assignment and other operators overloading, friend classes and

functions.

Week 4 Chapter 14 More About Classes: Static member, friend classes and functions, assignment and other

operator overloading, inner classes and nested classes.

Week 5, 6 Chapter 15 Inheritance, Polymorphism, and Virtual Functions

Week 7 Chapter 16 Exceptions, Templates, and the Standard Template Library

Week 8: Chapter 17 Linked Lists

Week 9 Chapter 18 Stacks and Queues

Week 10 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]