**CMPS 3500 Programming**

**Catalog Description**

**CMPS 3500 Programming Languages (3)**

An examination of underlying concepts in high level programming languages and techniques for the implementation of a representative sample of such languages with regard to considerations such as typing, block structure, scope, recursion, procedures invocation, context, binding, and modularity. Features of OOP, thread, synchronization and concurrency, functional function will be discussed. Prerequisite: CMPS 2020.

**Prerequisites by Topic**

Students are assumed to know at least one language very well including typing, program structure, subprogram and block structures.

**Units and Contact Time**

3 semester units. 2 units lecture (100 minutes), 1 unit lab (150 minutes).

**Type**

Required for CS

**Required Textbook**

Concepts of Programming Languages 9th edition by R. Sebesta, Published By Addison-Wesley ISBN 978-0-13-607347-5

**Recommended Textbook and Other Supplemental Materials**

1. Java API at <http://docs.oracle.com/javase/7/docs/api/>
2. Java Language Specification <http://docs.oracle.com/javase/specs/jls/se7/html/jls-0-preface7.html>
3. Any book on C# in CSUB library

**Coordinator(s)**

Huaqing Wang

**Student Learning Outcomes**

ACM/IEEE Body of Knowledge Topics:

Concurrency need to be added in

**Overview of programming languages *(PL1):***History of programming languages; Brief survey of programming paradigms; Procedural languages; Object-oriented languages; Functional languages.

**Virtual machines *(PL2):***The concept of a virtual machine; Hierarchy of virtual machines; Intermediate languages.

**Introduction to language translation *(PL3):***Comparison of interpreters and compilers; Language translation phases (context-free grammars **†**, static semantics and dynamic semantics **†,**lexical analysis, parsing).

**Declarations and types *(PL4):***The conception of types as a set of values with together with a set of operations; Declaration models (binding, visibility, scope, and lifetime); Overview of type-checking.

**Abstraction mechanisms *(PL5):***Procedures, functions, and iterators as abstraction mechanisms;Parameterization mechanisms (reference vs. value, pointers **†**); Expression and assignment statements **†**.

**Object-oriented programming *(PL6):***Object-oriented design; Encapsulation and information-hiding; Separation of behavior and implementation; Classes and subclasses; Inheritance (overriding, dynamic dispatch); Polymorphism (subtype polymorphism vs. inheritance); Class hierarchies.

Introduction to Concurrency (PL): Reasons, levels of concurrencies, controls, threads and synchronization.

**Functional programming (PL7):**Overview and motivation of functional languages

**† Topics not included in ACM curricula**

**ABET Outcome Coverage**

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

3h. Recognition of the need for and an ability to engage in continuing professional development.

3i. An ability to use current techniques, skills, and tools necessary for computing practice.

comprehension of the tradeoﬀs involved in design choices.

**Lecture Topics and Rough Schedule**

Week 1 Chapter 1 Preliminaries

Week 2 Chapter 2 Evolution of the Major Programming Languages

Week 3 Chapter 3 Describing Syntax and Semantics

Week 4 Chapter 4 Lexical and Syntax Analysis

Week 5 Chapter 5 Names, Bindings, and Scopes

Week 6 Chapter 6 Data Types

Week 7 Chapter 7 Expressions and Assignment Statements

Week 8 Chapter 8 Statement-Level Control Structures

Week 9 Chapter 9 Subprograms

Week 10 Chapter 10 Implementing Subprograms

Week 11 Chapter 11 Abstract Data Types and Encapsulation Constructs

Week 12 Chapter 12 Support for Object-Oriented

Week 13 Chapter 13 Concurrency

Week 14 Chapter 14 Exception Handling and Event Handling

Week 15 Chapter 15 Functional Programming Languages

**Design Content Description**

None

**Prepared By**

Huaqing Wang on [date]

**Approval**

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