**CMPS 350 Programming Languages**

**Catalog Description**

**CMPS 350 Programming Languages (5)**

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. Each Week lecture meets for 200 minutes and lab meets for 150 minutes. Prerequisite: CMPS 223.

**Prerequisites by Topic**

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

**Units and Contact Time**

5 quarter units. 4 units lecture (200 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**

None

 **Coordinator(s)**

Huaqing Wang

**Student Learning Outcomes**

ACM/IEEE Body of Knowledge Topics:

Concurrency is covered in chapter 13 and should be added in here.

**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**

Chapter 1 Preliminaries

Chapter 2 Evolution of the Major Programming Languages

Chapter 3 Describing Syntax and Semantics

Chapter 4 Lexical and Syntax Analysis

Chapter 5 Names, Bindings, and Scopes

Chapter 6 Data Types

Chapter 7 Expressions and Assignment Statements

Chapter 8 Statement-Level Control Structures

Chapter 9 Subprograms

Chapter 10 Implementing Subprograms

Chapter 11 Abstract Data Types and Encapsulation Constructs

Chapter 12 Support for Object-Oriented

Chapter 13 Concurrency

Chapter 14 Exception Handling and Event Handling

Chapter 15 Functional Programming Languages

Week 1 Chapter 1 and 2-introduction of programming languages and criteria of evaluating programming

 languages.

Week 2 Chapter 3 Syntax description and semantic and Lexical and Syntax Analysis

Week 3 Chapter 5 and 6 - Names, binding, scopes and data types.

Week 4 Chapter 7 & 8 - Expression and assignment statements and Statement Level Controls

Week 5, 6 Chapter 9 & 10 Subprograms and Implementations

 Week 7 Chapter 11 & 12 Abstract data types, generic data types, and Object-Oriented programming.

Week 8 Chapter 13- Introduction to concurrency computing, level of concurrency, threads and synchronization.

Week 9 Chapter 14 and 15 – Exceptional handling and Event-handling.

Week 10 Functional Programming Languages

**Design Content Description**

None

**Prepared By**

Huaqing Wang on [date]

**Approval**

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