

1. Course number and name

CMP 321 – Programming Languages

2. Credits and contact hours

3 credit hours, 5 contact hours

3. Instructor's or course coordinator's name

Dr. Michel Pasquier

4. Textbook, title, author, and year

R. Sebesta, *Concepts of Programming Languages*, 11th edition, Addison-Wesley, 2016.

Other supplemental materials

Development tools and docs for Python, Java, Scheme, Haskell, Prolog, LeX, etc.

5. Specific course information

a. Brief description of content of the course (catalog description)

Introduces the fundamental principles and techniques in the design and implementation of modern programming languages. Covers key topics such as syntax and semantics, binding and scope, data types, control structures, and expressions. Discusses different programming paradigms, such as imperative, functional, logic, and object-oriented.

b. Prerequisites or co-requisites

Prerequisites: CMP 256 (GUI Design and Programming) or COE 312 (Software Design for Engineers), and CMP 305 (Data Structures and Algorithms)

c. Indicate whether a required, elective, or selected elective course in the program

Required

6. Specific goals for the course

a. Specific outcomes of instruction

Upon completion of the course, students will be able to:

1. Describe and compare the readability, writability, reliability and cost factors of programming languages.
2. Define syntax using Backus–Naur Form (BNF) and Extended BNF.
3. Specify semantics using attribute grammar and axiomatic semantics.
4. Design and implement a lexical analyzer and a recursive-descent parser for the EBNF of a simple language.
5. Analyze and evaluate different approaches to naming, storage binding, scopes, data types, control flow, and expressions.
6. Describe how the computer system uses activation records to manage program modules and their data.
7. Explain the characteristics of functions and lambda expressions and apply them to develop simple programs using functional programming.
8. Describe the components and mechanisms of predicate calculus and apply them to develop simple programs using logic programming.

b. Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course

This course contributes in a significant way to the accomplishment of the following program outcomes:

Program outcome	Emphasis in this course
(1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.	
(2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.	●
(3) Communicate effectively in a variety of professional contexts.	
(4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
(5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.	○
(6) Apply computer science theory and software development fundamentals to produce computing-based solutions.	●

Emphasis: ● High; ● Medium; ○ Low; Blank – Nothing Specific Expected

7. Brief list of topics to be covered

- i. Introduction to programming languages
- ii. Describing Syntax and Semantics
- iii. Lexical analysis and Syntax parsing
- iv. Names, bindings, and scopes
- v. Data types and control structures
- vi. Expressions, assignments, and regular expressions
- vii. Subprograms, run-time stacks, and activations records
- viii. Functional programming
- ix. Logic programming