COMP 3220, Principles of Programming Languages

Credit hours: 3 lecture
Contact hours: 3 lecture

Catalog Description: Study of programming language principles supporting procedural abstraction, data abstraction, storage allocation, and parallel execution; language types and examples; language translations.

Prerequisites: COMP 2210
Corequisites: None

Required Course (CSCI, SWEN)

Instructor or Course Coordinator: Dr. Jeff Overbey

Required Textbook

Course Outcomes
The student will be able to
- understand the concepts and paradigms associated with programming languages.
- critically evaluate programming languages.
- understand syntactic definition issues, starting with tokens and regular expressions and the use of BNF and its relationship to grammars and language translation.
- understand language semantic issues associated with traditional imperative programming languages, declarations and typing, control flow, procedures and parameters.
- understand language abstraction mechanisms, classes, modules and abstract data types.
- understand different language paradigms (object-oriented, functional, imperative, logic) and their theoretical foundations.

Topics Covered
- Introductions, Language history, paradigms, definition, translation and design (3 hours)
- Programming Language syntax and semantics (4 hours)
- Logic Programming (3 hours)
- Basic Language Principles, names, scopes, storage (4 hours)
- Functional Programming (4 hours)
- Data types and polymorphism (3 hours)
- Language Control and abstraction: Expressions, Statements, Functions and Procedures (8 hours)
- Language Implementation (6 hours)
- Object Oriented Programming (5 hours)
- Storage abstraction: Classes, Abstract Data Types and Modules (3 hours)
- Exams (2 hours)
Course Requirements

- Homework Assignments (50%)
- Midterm Exam (20%)
- Final Exam (30%)

Syllabus prepared: Spring 2016