COMP 333 Fall 2013

Concepts of Programming Languages

Class Section Number: 15021

Meets: TTH 9:30 – 10:45 in JD 3508

INSTRUCTOR: D. Schwartz

# OFFICE: JD 4401

EMAIL: [diane.schwartz@csun.edu](mailto:diane.schwartz@csun.edu)

WEB PAGE: www.csun.edu/~dls

OFFICE HOURS: TBD

PREREQUISITES: Comp 122/L (C or better), 182/L (C or better)

COURSE DESCRIPTION: Discussion of issues in the design, implementation, and use of high-level programming languages. Historical background. How languages reflect different design philosophies and user requirements. Technical issues in the design of major imperative (procedural) programming languages. Other approaches to programming: functional programming, logic programming, and object-oriented programming and concurrency.

REQUIRED TEXTBOOK: Programming Language Pragmatics (Third Edition)

by Michael L Scott, Published by Morgan Kaufmann

GRADING: Midterm Exams (2) 100 pts each

Final Exam 150 pts

Homework/Programming Assignments 150 pts weighted total

Plus and minus grading will be used. Your grade will be based on the sum of the points you receive on all of the exams and homework/ programming assignments listed above. There is a total of 500 pts.

COURSE OBJECTIVES

A successful student will be able to:

1. Summarize the evolution of programming languages illustrating how this history has led to the paradigms available today.
2. Explain the models of translating high level languages to machine language, including the phases and files in the compilation process.
3. Demonstrate the use of BNF to describe concrete syntax, and to apply the BNF structures to translation of programs.
4. Demonstrate different forms of binding, visibility, scoping, and lifetime management.
5. Explain the importance of types and type-checking in providing abstraction and safety.
6. Evaluate the tradeoffs between the different paradigms, considering such issues as space efficiency, time efficiency, safety, and power of expression.
7. Design, code, test, and debug basic programs using the functional, logical, and concurrent paradigm
8. Explain the communication and synchronization models of concurrent processes.
9. Explain the importance of abstractions, especially how abstraction mechanisms support the creation of reusable software components.

(Continues on next page)

COURSE OUTLINE (Tentative)

|  |  |  |
| --- | --- | --- |
| **Weeks** | **Lecture Topics** | **Text Reading Assignment**  **from the Scott text** |
| 8/27, 8/29 | Introduction | Chap 1 |
| 9/3 /9/5 | Programming Language Syntax | Chap 2 |
| 9/10, 9/12 | Syntax Analysis | Chap 2 ( thru 2.3.2) |
| 9/17, 9/19 | Functional Programming (Scheme) | Chap 10 |
| 9/24, 9/26, 10/1 | Functional Programming (Scheme) | Chap 10 |
| **Thursday 10/3** | **Midterm 1** |  |
| 10/8, 10/10 | Names, Bindings and Scope | Chap 3 |
| 10/15, 10/17 | Expressions and Control Structures | Chap 6 |
| 10/22, 10/24 | Data types and Type checking | Chap 7 |
| 10/29, 10/31, 11/5 | Subprograms and Parameter Passage | Chap 8 |
| 11/5, 11/7 | Data Abstraction and OOP ( Java) | Chap 9 |
| **Tuesday 11/12** | **Midterm 2** |  |
| 11/14, 11/19, 11/21 | Logic Programming ( Prolog) | Chap 11 |
| 11/26 | Logic Programming (Prolog) | Chap 11 |
| 12/3, 12/5 | Concurrent Programming (in Java) | Chap 12 |
| 12/10 | Review Day |  |
| **12/17 Final Exam**  **8am – 10am** | **Final Exam covers material from the entire course** |  |

**See the class Moodle page for homework and programming assignment postings and other information.**