

INSTRUCTOR: Professor Diane Schwartz
OFFICE: JD 4401
EMAIL: diane.schwartz@csun.edu
MOODLE SITE: moodle.csun.edu
OFFICE HOURS: TTH 9:30 – 10:30

Section: 15313 Meets TTH 8:00-9:15 in JD 3508
Section: 15792 Meets TTH 11:00-12:15 in JD 3508

PREREQUISITES: Comp 282 (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 (3)	75% (25% each)
	Homework/Programming Assignments	20%
	Class Participation	5%
	(Attendance, Classroom Activities)	

Grades will be based on percentage of total points earned. Plus and minus grading will be used.
A-, A : 90 – 100%; B -, B, B+: 80 – 89%; C -, C, C+: 65 – 79%; D: 60 – 64%; F: < 60%

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.

Additional Class Information

1. Class attendance/participation counts for 25 points and will be scaled to 5% of your grade. You will get the full 25 points if you miss no more than 3 class meetings. Absences beyond 3 class meetings, will cause you to lose 5 pts per missed class. You will not lose more than 25 points total. To be counted as attending class you need to sign the attendance sheet at the beginning of class, participate in the classroom activities and remain in class until the end of class.

2. Students are expected to read and abide by the University's Academic Honesty statement printed in the current catalog. Academic dishonesty will result in a zero on the assignment and can result in class failure.
3. Some of the programming assignments will be done in teams of 2 or 3 students. Sometimes your instructor will assign the teams and sometimes you will be able to select your own team.
4. There are no make-up exams.
5. Lecture power points will be posted on Moodle.

Class Etiquette

1. No use of cell phones, lap tops or other communication/computer devices is allowed. You should have your full attention on the class.
2. Please do not eat in class. Don't bring in breakfast, lunch, dinner or snacks to eat in class. Bottles of water, small sodas or cups of coffee or tea are ok.
3. Come to class on time and do not leave early. The attendance sign-up sheet will be available only at the beginning of class.
4. Full participation in class activities is expected.

COURSE OUTLINE (Tentative)

Lecture Days	Lecture Topics	Text Reading Assignment from the Scott text
1/26, 1/28	Introduction and History of Programming Languages	Chap 1.1 – 1.3
2/2, 2/4	Programming Language Syntax	Chap 2.1
2/9,2/11	Introduction to Compilation	Chap 1.4 – 1.6
2/16,2/18	Names, Scopes and Bindings	Chap 3.1, 3.2, 3.3
2/23,2/25	Data Types and Type Checking	Chap 6.1, 7.1,7.2,7.3, 7.7.3, 7.10
3/1	Subroutines	Chap 8.1, 8.3, 8.5
March 3	Midterm 1	
3/8, 3/10	Functional Programming (Scheme)	Chap 10
3/15, 3/17	Functional Programming (Scheme)	Chap 10
3/22, 3/24	SPRING BREAK	
3/29	Functional Programming (Scheme)	Chap 10
3/31	CESAR CHAVEZ DAY	
4/5	Functional Programming (Scheme)	Chap 10
April 7	Midterm 2	
4/12, 4/14	Logic Programming (Prolog)	Chap 11
4/19, 4/21	Logic Programming (Prolog)	Chap 11
4/26, 4/28	Concurrent Programming (in Java)	Chap 12
5/3, 5/5	Concurrent Programming (in Java)	Chap 12
5/10	Catch-up or Review	
May 12	Midterm 3	
No Final	No final exam	

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