

# Lectures

**Classroom:**

Engineering Building 110

**Time:**

Mon, Wed: 7:25 – 8:50p.

(extended to 9:15p if any classes are missed)

2 - 3 minute break approximately halfway  
through each class

# Contact Info

## Instructor

**Name:** Zerksis D. Umrigar  
**Office Hours:** Mon, Wed: 6:20 – 7:20p  
**Office:** EB P14  
**Phone:** (607) 777-4316 during office hours  
**Email:** umrigar@binghamton.edu

Out of town on Thurs & Fri, but will read email.

## Teaching Assistants

<b>Name:</b>	Abhishek Jain	John Weachock
<b>Office Hours:</b>	Mon, Wed: 1:00 – 2:00p	Tue, Thu: 1:30 – 2:30p
<b>Office:</b>	EB G25	EB P17
<b>Email:</b>	ajain13@binghamton.edu	jweacho1@binghamton.edu
<b>Responsibilities:</b>	Project & quiz grading	Homework, midterm & final grading

# Grading

<b>Pop Quizzes:</b>	15% If a total $n$ quizzes are given, then $\lfloor n/3 \rfloor$ lowest quiz grades dropped.
<b>Homeworks:</b>	25% Lowest homework grade dropped
<b>Projects:</b>	25% Lowest project grade dropped
<b>Midterm:</b>	15%
<b>Final:</b>	20%

Quizzes closed book. Final, midterm open-book, open-notes; no electronic devices.

Projects and homeworks may be submitted up to 1 week late with a 15% penalty.

# Letter Grade Assignment

- Letter grades assigned strictly monotonically based on numeric course grade.
- A letter grade of A will be given only for consistent superior work.
- It should be relatively easy to get a grade around a B.
- You will get an F only if you miss turning in a lot of work or submit consistently very poor quality work or if you cheat.
- TA Grading Guidelines are available.

# Academic Honesty

Cheating of any type will be penalized heavily.

- Minimal penalty: an F letter grade for entire course.
- Permissible to collaborate to understand course material, homework questions or project assignments. Not permissible to discuss solutions.
- If you feel you may have inadvertently crossed the line, then let us know; will not be considered cheating.
- If submitting an assignment late after solution has been posted, you should obviously not be looking at the solution.
- All registered students must sign and complete an Academic Honesty Statement.

# Texts

## Required Text

Michael L. Scott, *Programming Language Pragmatics*, Fourth Edition, Morgan-Kauffman, 2016.

Since midterm and final will be open-book, open-notes but no electronic devices will be permitted, it is probably a good idea to purchase hardcopy.

# Course Web Site

- All course material on course web site at `http://zdu.binghamton.edu/cs571-16f`.
- Course web site mirrored at `http://cs.binghamton.edu/~umrigar/cs571-16f`.
- Slides usually available by 6:00p before class.
- Course web site available via `git` repository at `ssh://user@remote.cs.binghamton.edu:/~umrigar/cs571-16f.git`. Useful for tracking changes.

# Miscellaneous

- All students registered for course should subscribe or monitor the blackboard discussion group.
- Students should bring a # 2 pencil and eraser to each class in order to fill-in the quiz grid sheets. **Ink is not acceptable.**
- Grades will be made available on blackboard.
- All graded material can be picked up from Mr. Weachock during his office hours.



# Objectives

- Obtain an understanding of semantics for C-like procedural languages, including memory allocation, parameter passing, and scope management.
- Familiarization with typical features of Java/C++-like object-oriented languages, such as heap management, messaging, inheritance, polymorphism, encapsulation.
- Acquire experience with Haskell-like functional languages and functional concepts such as lists, tuples, first-class functions, and map/filter/fold.
- Understand programming and semantics for Prolog-like logic languages, such as facts and rules, derivations, and tabling.
- Obtain an overview of compiler construction, including phases such as lexical analysis, parsing, semantic analysis, optimization, and code generation.

# Prerequisites

- Familiarity with an object-oriented language like Java.
- Basic data-structures and algorithms.
- Some level of mathematical maturity.
- Comfort within a Unix environment.

# Topics Covered

- Programming languages overview.
- Language specification: lexical, syntactic, semantic.
- Names.
- Control flow.
- Types.
- Subroutines.
- Object-oriented programming.
- Functional programming.
- Logic programming.
- Concurrent programming.
- Scripting languages.
- Language eco-systems.

Emphasis will be on alternate programming paradigms.