EECS 368 Programming Language Paradigms

David O. Johnson Fall 2022

Who Am I

- Dr. David O. Johnson
- BSEE, Kansas State University, 1975
- MSEE, Kansas State University, 1976
- PhD Computer Science, University of Kansas, May 2009
 - Dissertation: Human Robot Interaction Through Semantic Integration of Multiple Modalities, Dialog Management, and Contexts
- Bell Labs, 1976-1980, member of technical staff, packet switching and international standards
- Uninet, 1980-1986, software engineer and manager, packet switching
- Adacom, 1986-1987, software engineer and manager, data communication equipment
- Digital Equipment Corporation (now Hewlett-Packard), 1987-1989, software sales consultant, mini-computers and data communication software
- Sprint, 1989-2007, manager and senior analyst, Sprint's internal voice and data communications networks

Who Am I

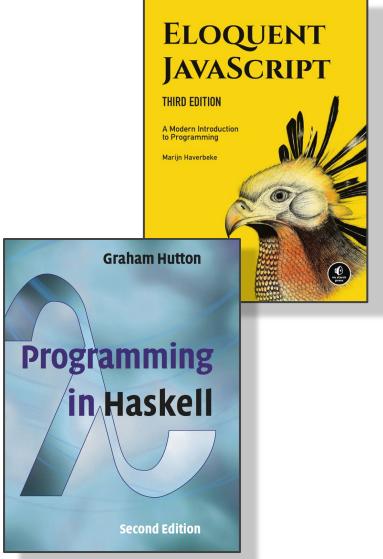
- University Missouri Kansas City, Spring 1985, 2009-2013, Adjunct EECS Instructor
- Eindhoven University of Technology, Netherlands, 2012-2013, Post-Doctoral Researcher
- Northern Arizona University, Flagstaff, AZ, 2014-2016, Post-Doctoral Researcher
- University of Kansas, 2016-present, Associate Teaching Professor
 - 140/141 Introduction to Digital Logic Design
 - 210 Discrete Structures
 - 316 Circuits, Electronics, and Instrumentation
 - 368 Programming Language Paradigms
 - 448 Software Engineering
 - 581/582 Computer Science Design I & II
 - 658 Introduction to Machine Learning
 - EE, CoE, and CS High School Summer Camp

Contact Information

- Instructor: Dr. David O. Johnson
- Phone: 913-461-4304
 - If you call, leave a message otherwise I won't return your call
 - Text me
- E-mail: davidojohnson@ku.edu
- Office: Eaton 2005D
- Office hours:
 - Thursday, 2:30-3:45
 - Friday, 12:00-2:45
 - By appointment, or whenever the light is on in my office

No Class Textbooks

- None required.
- All lecture slides will be available on Canvas.
- Any supplemental information will also be available on Canvas.
- Everything is online
- However, if you do need a textbook, then
 - Haverbeke, Eloquent JavaScript
 - https://eloquentjavascript.net/
 - Hutton, Programming in Haskell
 - http://www.cs.nott.ac.uk/~pszgmh /pih.html



Catalog Description

- The course is a survey of programming languages: their attributes, uses, advantages, and disadvantages.
- Topics include scopes, parameter passing, storage management, control flow, exception handling, encapsulation and modularization mechanism, reusability through genericity and inheritance, and type systems.
- In particular, several different languages will be studied which exemplify different language philosophies (e.g., procedural, functional, object-oriented, logic, scripting).
- Prerequisite: EECS 268 and upper-level EECS eligibility.

Course Topics

- Evolution of Programming Languages
 - Rationale behind programming languages
 - Programming languages in context
- Programming Language Processing
 - Compilation
 - Interpretation
 - Mixed approaches
- Programming Language Characteristics
 - Imperative, functional and declarative languages
 - Parameter passing and evaluation order
 - Iteration, recursion, and continuation
 - Domain Specific Languages (DSLs)

- Basics of JavaScript Programming
 - Call-by-value evaluation
 - Dynamic typing
 - Iteration and concurrency
- Basics of Cloud Computing
 - Web services
 - Client/server
 - Synchronous vs. asynchronous programming
 - Building systems with high availability
 - Programming at scale
- Basics of Haskell programming
 - Lazy evaluation
 - Early binding and type inference
 - Parametric polymorphism
 - Constructed types and pattern matching
 - Type classes and instances
 - Functors and semantic algebras

Grading Policy

• A: 90-100%

• B: 80-89%

• C: 70-79%

• D: 60-69%

• F: Below 60%

All grades are rounded to nearest whole integer (e.g., 89.5 → 90, but 89.4999... → 89)

Grade Composition

- In-Class Problems: 50%
- Programming Assignments: 50%
- No Exams!

In-Class Problems

- Lectures will include problems to be completed during class and handed in by the end of the day.
- You must submit your completed In-Class Problem as a PDF to Canvas (under In-Class Problem Submissions) before 11:59 PM
 - See PDF Help file on Canvas under In-Class Problem Submissions.
- You may work with other students in the class to solve the problems.
- Two Supplemental Instructors and I will also be available to help during class and office hours.
 - Abhishek Doodgaon
 - Soujanya Ambati
- Make ups for In-Class Problems are allowed for excused absences only!

In-Class Problem Grading

- 90-100 points
 - Show your work
 - Correct answer
- 80-90 points
 - Show your work
 - Incorrect answer due to a minor error (e.g., math error)
- 70-80 points
 - Show your work
 - Incorrect answer due to a major error (e.g., incorrect formula, didn't finish solution)
- 60-70 points
 - Show your work
 - No answer
- 60 points
 - Don't show your work
 - Correct answer
- 0 points
 - Otherwise

- If the problem has multiple parts, each part is graded separately and averaged.
- Caveat:
 - It is hard to "show your work" for some simple problems
 - Lenient on what constitutes "show your work"

Programming Assignments

- 7 during the semester
- Your Programming Assignments are individual-effort.
- All deliverables for a programming assignment should be submitted in a single zip file to Canvas by 11:59 PM on the due date.
- Late programming assignments are NOT accepted, except for an excused absence.

Help on Programming Assignments

- Your classmates
 - You can brainstorm with other students and help them work through problems in their programs, but everyone should have their own unique assignment programs.
- Supplemental Instructors
 - Office hours will be posted on Canvas
- Me
 - Via email
 - During office hours
- Internet
 - Yes, you may use the Internet for help!
 - Why? Because that's what professional programmers do!
 - stackoverflow (stackoverflow.com) is a great to place to start.
 - Google is a great place to find new places, too.

Programming Assignment Comments

- Two types of comments will be required on all EECS 368 programming assignments:
 - Prologue Comments
 - In-Line Comments
- Why?
 - Reducing the cost of software maintenance is the most often cited reason for commenting programs.
 - 40%–80% of the lifetime cost of a piece of software goes to maintenance.
 - Hardly any software is maintained for its whole life by the original author.
 - Commenting improves the readability of the software, allowing engineers to understand new code more quickly and thoroughly.

Comments

Prologue Comments

- Name of code artifact
- Brief description of what the code does
- Programmer's name
- Date the code was created
- Preconditions
 - Acceptable and unacceptable input values or types, and their meanings
- Postconditions
 - Return values or types, and their meanings
- Error and exception condition values or types that can occur, and their meanings

In-Line Comments

- What code does, is not "intuitively" obvious
- Comments help maintenance programmers
- Comments also help <u>you</u> remember why you coded something the way you did
- Best practices:
 - Between major blocks of code
 - Every line

Comments Rubric

Exceeds Expectations	Meets Expectations	Unsatisfactory	
(90-100%)	(80-89%)	(0-79%)	
Software is adequately	Prologue comments are	Prologue comments are	
commented with prologue	present but missing some	missing all together or there	
comments, comments	items or some major blocks of	are no comments on major	
summarizing major blocks of	code are not commented or	blocks of code or there are	
code, and comments on every	there are inadequate	very few comments on each	
line.	comments on each line.	line.	

Assignment Grading

- We will strive to have the assignments graded one week after they are turned in.
- Solutions to the In-Class Problems will be covered in the following lecture.
- There are two graders for the class:
 - Anjali Roy (email: <u>a820r843@ku.edu</u>)
 - Charan Suresh (email: <u>charansuresh@ku.edu</u>)
- They will take turns grading the In-Class Problems and Programming Assignments as shown in the Course Schedule.
- If you have questions about the grading, start with the graders.
- Then, if you are not satisfied, see me.
 - Note: I rarely over-turn a grader's grading.

Course Schedule

Date	Day	Lecture	Chapters	Assignment Due	Grader
8/22/2022	М	Class Introduction			
8/24/2022	W	Overview of Programming Languages			Anjali
8/26/2022	F	JavaScript Values, Types, and Operators	Introduction & 1		
8/29/2022	М	JavaScript Program Structure	2		
8/31/2022	W	JavaScript Functions	3		Charan
9/2/2022	F	JavaScript Objects and Arrays	4		
9/5/2022	М	Labor Day			
9/7/2022	W	JavaScript Higher-Order Functions	5	1 (Anjali)	Anjali
9/9/2022	F	The Secret Life of JavaScript Objects	6		
9/12/2022	М	JavaScript Bugs and Errors	8		
9/14/2022	W	JavaScript and the Browser	13		Charan
9/16/2022	F	JavaScript and the DOM	14		
9/19/2022	М	Handling Events in JavaScript	11 & 15	2 (Charan)	
9/21/2022	W	JavaScript SVG & Canvas	17 (pp. 295-300)		Anjali
9/23/2022	F	More on JavaScript Canvas	17 (pp. 300-306)		l
9/26/2022	М	Even More on JavaScript Canvas	17 (pp. 306-310)		
9/28/2022	W	НТТР	18 (pp. 319-323)		Charan
9/30/2022	F	JavaScript & HTTP	18 (pp. 323-326)		
10/3/2022	М	JavaScript & HTML Forms	18 (pp. 326-330)	3 (Anjali)	
10/5/2022	W	More JavaScript & HTML Forms	18 (pp. 330-335)		Anjali
10/7/2022	F	Even More JavaScript & HTML Forms	18 (pp. 335-340)		
10/10/2022	М	Fall Break			
10/12/2022	W	JavaScript & Node.js	20 (pp.363-368)		Charan
10/14/2022	F	JS Node File System & HTTP	20 (pp.369-372)		

JavaScript Text: Eloquent JavaScript, https://eloquentjavascript.net/

Haskell Text: Programming in Haskell, http://www.cs.nott.ac.uk/~pszgmh/pih.html

Course Schedule

Date	Day	Lecture	Chapters	Assignment Due	Grader	
10/17/2022	Μ	JS Node File Server	20 (pp.372-378)	4 (Charan)		
10/19/2022	V	JS Node File Server (con't)	20 (pp.378-380)		Anjali	
10/21/2022	F	Application Programming Interfaces (API)				
10/24/2022	М	Introduction to Haskell	1			
10/26/2022 W		Basics of Cloud Computing (Nick Smith)			Charan	
10/28/2022	F	Haskell First Steps	2			
10/31/2022	М	Haskell Types and Classes	3	5 (Anjali)		
11/2/2022	W	Defining Haskell Functions	4		Anjali	
11/4/2022	F	Haskell List Comprehension	5			
11/7/2022	М	Recursive Haskell Functions	6			
11/9/2022	W	Higher-Order Haskell Functions	7		Charan	
11/11/2022	F	Declaring Haskell Types	8			
11/14/2022	М	Haskell Countdown Example	9	6 (Charan)		
11/16/2022	W	Haskell Interactive Programming	10		Anjali	
11/18/2022	F	Haskell Lazy Evaluation	15		ľ	
11/21/2022	М	No Class				
11/23/2022	V	Thanksgiving Break				
11/25/2022	F	Thanksgiving Break				
11/28/2022	Μ	Haskell Monads	12.3			
11/30/2022	W	Haskell Sudoku Example (Part 1)			Charan	
12/2/2022	F	Haskell Sudoku Example (Part 2)				
12/5/2022	М	Haskell QuickCheck			Anjali	
12/7/2022	W	More on Haskell QuickCheck		7 (Anjali)		
12/9/2022	F	Stop Day				

JavaScript Text: Eloquent JavaScript, https://eloquentjavascript.net/
Haskell Text: Programming in Haskell, http://www.cs.nott.ac.uk/~pszgmh/pih.html

Canvas

Home/Modules

- Syllabus & Class Introduction
- Office Hours & Contact Information
- Course Schedule (with due dates)
- Lectures (available prior to class)
- In-Class Problem Submissions
 - PDF Help
- Programming Assignments

Announcements

Grades

Excused Absence Policy

- Medical
 - Doctor's note covering date of absence
- Job Interview
 - Documentation of interview from company
- Bereavement
 - See http://policy.ku.edu/human-resources/bereavement-leave
 - Documentation:
 - US citizens: obituary
 - International students: official visa papers
- University Sponsored Activities (e.g., sports, band)
 - Email from activity sponsor covering date of absence
- Others
 - See me

Student Access Center (fka AAAC)

- The Student Access Center assists students with disabilities by facilitating accommodations that remove barriers to their academic success.
- They authorize reasonable and appropriate accommodations for qualified students with documented disabilities.
- If you have a disability for which you wish to request accommodations and have not contacted the SAC, please do so as soon as possible.
- Their phone number is 785-864-4064 (V/TTY).
- Their email is: access@ku.edu
- Their office is located in 22 Strong Hall
- Information about their services can be found at: <u>access.ku.edu</u>.

Academic Misconduct

- The EECS Department regards academic misconduct as a very serious matter.
- Students who violate conduct policies will be subject to severe penalties, up through and including dismissal from the School of Engineering.
- Please refer to the KU Policy Library website under Academic at http://www.policy.ku.edu/ for specific guidelines about actions considered to be academic misconduct and the repercussions of such action.
- These actions include, but are not limited to:
 - Disruption of classes
 - Threatening an instructor or fellow student in an academic setting
 - Giving or receiving unauthorized aid on examinations or in preparation of notebooks, themes, reports or other assignments
 - Knowingly misrepresenting the source of any academic work
 - Unauthorized changing of grades
 - Unauthorized use of University approvals or forging of signatures
 - Falsification of research results
 - Plagiarizing of another's work
 - Violation of regulations or ethical codes for the treatment of human and animal subjects
 - Or otherwise acting dishonestly in research

Any Questions?