Week	Tuesday	Thursday
2 **Feb 4-8**	Functional vs. imperative programming languages Functions, lambda functions, functional forms Side effects and referential transparency LISP lists and data representation ceclnx01.cec.miamioh.edu • bash reference (might need to be on-campus) • See `~zmudam/CSE465-565/readme` • `cp -r ~zmudam/CSE465-565/ .` `basics.scm` • Arithmetic expressions • Function definitions • Conditionals - if and cond Readings: Sections 15.1-15.5, notebook pages **1-11**	Quiz #1 on Tue Feb 12 • Chapter 1 - 35% • Chapter 2 - 20% • Chapter 15.1-15.5 - 45% conditionals quote car, cdr caddar, cadaar, …. cons, list, append recursion as control structure for repetition `list.scm`
1 **Jan 28-Feb 1**	Canvas and Google Docs Syllabus Course notebook Chapter 1 • Compiled vs interpreted languages Criteria for evaluation of language features Homework #1 (due Mon Feb 11 11:59pm) Readings: Chapter 1	Need course notebook next week (or first 11 pages) Any questions on homework #1? Chapter 1 (continued) Language survey Short history of PLs (author's slides) • Evolution of programming languages Readings: Chapter 2
IGNORE STUFF BELOW. STAGING AREA FOR FUTURE CLASSES		

3
Feb 11-15
	Quiz #1
	`mystery.scm`
	`equal.scm`
	• `=, EQ?, EQUAL?, EQV?`
	Discuss Homework #2
		`tail.scm`
		`code.scm`
		• `eval, apply, lambda, map`
		• `eval`
		  ○ `(interaction-environment) lnx01`
		  ○ `user-initial-environment mit`
		• `On lnx01, need:`
		  ○ `,open random`

		`imperative.scm` ● Imperative features in Scheme Questions on HW2?
14 **May 1**	Final exam topics (topics finalized on Wednesday) Semantics ● Denotational semantics (75-76) ● Operational semantics ● Axiomatic semantics Readings: Section 3.5	Semantics ● Axiomatic semantics HW5 not graded yet ● Test cases (FYI) Review for final exam ● Email me request to switch exam dates
13 **Apr 24**	Quiz #5 ● IEnumerable/IEnumerator ● Chapter 9 ● Grammars Discuss Homework #6 Ambiguity continued (73) Describing Syntax ● Conveniences of {} and [] ● Sample Grammars ● Java ● SQL ● Pascal grammar (70-71, 72) ● Attribute grammar (74) Readings: Section 3.4	Quiz 5 Post-semester outcomes survey ● Complete before Fri May 5 11:59pm. Part of HW6 Semantics ● Denotational semantics (75-76) ● Operational semantics ● Axiomatic semantics Readings: Section 3.5
12 **Apr 17**	Subprograms/funcions ● Procedures, functions, subprograms, and Methods ● Compare with yield statement (59) ● Call by value, reference, value-result, name ● C++, Java, Python, C# revisited (63) ● Named parameters and default parameters (62) ○ pythonNamedParameters.py ● Variadic functions - varargs.{cpp, py, cs, java}	Functional tools in imperative languages (46-50) ○ Python lambda expressions, map ○ C# delegates, select, where ○ C++ lambda expressions Closures and Coroutines (author Chp. 9 slides 46-52) Javascript example (67) Describing Syntax ● Grammars/BNF ● Derivations and parse trees (73)

	(68-69) ● Generics/templates ○ C# samples ● Macros (64) Readings: Chapter 9	● Ambiguity (73)

11
Apr 10
	Quiz #4
	● C++ pointers and dynamic memory
	● Smart pointers
	● Using dynamic memory within class definition
	● Array access functions
	Data Types
	● Subranges
	● Associative arrays
	● C union - union.cpp
	● Type checking, strong type checking
	Expressions and assignment statements
	● Precedence and associativity
	● Evaluation order
	● Side effects
	● Operator overloading
	● Narrowing, widening conversions, casting (56-57)
	● Boolean expressions, short circuiting
	● Compound assignment, unary assignment
	● Assignment as an expressions
	Readings: Chapter 6 & 7
		Quiz #4
		Statement-level control structures
		● C# enumeration (58-59)
		● C++ iterators (60-61)
		● if ()
		if ()
		else
		● Switch C++, C#, Ada
		● Logically controlled loops
		● Traditional counter controlled loop
		● Goto
		● break and continue
		Subprograms/functions
		● Multi-dimensional arrays in C++
		○ void f(int mat[50][100]) vs
		○ void f(int mat[][100]) vs
		○ void f(int mat[50][])
		Readings: Chapters 8 & 9

10 **Apr 3**	Smart pointers in C++ (42) ○ smartPtr.cpp ○ unique_ptr and shared_ptr C++ class definitions and memory management ● mystring.h and main.cpp (43-45) ● `-fno-elide-constructors` compile option Data types ● Multi-dimensional arrays, row major vs column major layout Readings: Chapter 6	Data types ● Arrays, slices, addressing ○ Revisit Arrays.cs (55) ● Reflection.cs (53-54) ● Managing the heap Discuss Homework #5

9 **Mar 27**	Quiz #3 on Wednesday ● C# ● Static/dynamic scope ● Chapter 5 Dynamic/static scope review ● Referencing Environment ● Python scoping (52) Names, bindings, type checking, scopes Static, stack dynamic, heap variables Lifetime PointersAndMemoryExamples.docx (37)	Quiz #3 ptr2.cpp (38-39) Memory allocation problems in C++ (memory.cpp 40-41) ● Lost memory ● Dangling pointers ● Deallocating item instead of array of items ● Deallocating incorrect memory ● Array overruns Heap diagnostic tool -- valgrind (~zmudam/lang/readme)
Spring Break	No class.	No class.
8 **Mar 13**	No class due to instructor illness.	Office hours canceled for today Discuss Homework #4 DataStructures - DataStructures.cs, complex.cs ● List ● Sets ● Dictionary ● Properties ● Templates/generics ● Classes Static & dynamic scoping Scope -- scope.py scope.js rhino Javascript implementation Readings: Chapter 5
	Midterm exam	Review exam Homework #4 to be distributed next week. C# ● C# reference manual ● C# API reference ● Safari references (links work on campus) ○ Essential C# 6.0

7 **Mar 6**		○ C# 6.0 in a Nutshell Arrays - arrays.cs ● Compiling ceclnx01 & MS VS ● Overview, syntax, basic output ● [][] vs [,] ● switch/case statement ParameterPassing - ParamsAndEnums.cs ● Parameter passing modes - in, out, and ref NamespaceClassAndStruct - NamespaceClassAndStruct.cs ● Namespaces ● struct vs. class
6 **Feb 27**	Logic and logic programming Python 3.6 ● Python reference manual ● Python API ● Safari references ○ Introducing Python ○ Learning Python Python basics First.py Python collection classes ● Lists, indexing ● Dictionary ● Set	Midterm exam on Mon Mar 6 Uninstantiation of Prolog variables when failure occurs Python continued ● fraction.py, problem.py, main.py ● io.py ● listComp.py ● List comprehensions ● In-class: define Bag class
5 **Feb 20**	Quiz #2 Prolog continued prolog on ceclnx01 ● `not(X=Y)  vs  X \= Y` ● `not(P(X)) vs \+ P(X)` list.pl ● length built-in ● [H|T], [H1,H2|T] ● reverse trace ● append trace trace and notrace	Office hours on Friday noon-1:00 kth parse.pl cut.pl game.pl Discuss Homework #3 Prolog practice ● sum(L, S) ✔ ● max(LST, E) ✔ ● lastElement(L, E) ✔ ● nextToLast(LST, E) ● kth(K, LST, E) ✔

		• zip(L1, L2, Z)

4
Feb 13
	HW1 notes
	List representation
	Prolog
	• basics.pl
	• relatives.pl
	Software
	• `ceclnx01 - gprolog`
	• SWI-Prolog
	
	Readings: Chapter 16
		HW2 - getOrder should return first one, if there are multiples
		Quiz #2 on Monday
		• Scheme 30%
		  ○ apply, eval, lambda, map
		  ○ tail recursion
		• Prolog 70%
		  ○ Reading Prolog and evaluating queries
		  ○ Writing Prolog
		  ○ List processing
		Prolog
		  • relatives.pl
		  ○ review uncle and uncle2 (relatives2.pl)
		  ○ ancestor trace
		  • speed.pl
		  ○ arithmetic
		  • list.pl
		• [H|T], [H1,H2|T], length built-in
		• Write: sum(L, S), lastElement(L, E), nextToLast(L, E), kth(K, L, E)
		• reverse trace
		• append trace

Final Exam Week	8:30 section - Tue May 14 8:00-10:00 1:15 section - Tue May 14 12:45-2:45	**Office hours for exam week** Mon - 1:00-2:00 Tue - 3:00-4:00 Wed - none Thr - none Fri - none