

Programming Language Paradigms

Course Number: EECS 368

Instructor: Dr. Andrew (Andy) Gill

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.

Textbook

None required

Grading

Scale and Weight: The course will be graded on a standard 10 point scale:

A - 90-100
B - 80-89
C - 70-79
D - 60-69
F - 0-59

Assessments will be made based on a mid-term exam, a final exam, as well as homeworks and/or project(s).

Class Participation

I do not take attendance in class; however participation in class is important to your success. Some of the material presented in class is not in any text. You must come to class to be successful. Please ask questions and participate in class discussions. When assigning final grades, borderline cases will be decided based on class participation.

Grading Errors

If I have made an error in grading an exam or assignment, you have two weeks following the date the item is available to see me about correcting the problem. Note that this includes the final! After that time, your grade is set and will not be changed. I also request that you wait 24 hours after an exam is returned before coming to me with questions.

Curving

I may decide to curve final scores when the semester is over. Whether I curve and how much I curve is at my discretion. I will never curve scores on an individual graded item.

Email

I encourage you to use email to contact me directly, and use piazza if you have general questions. I am logged in when I am in the building and check my mail frequently. Email is my preferred means of communication, thus I will frequently send email to your KU account, so please check it regularly. I send official class information to only your KU account.

Office Hours

I will make every effort to be in my office during scheduled office hours. If there are exceptions, I will let you know as early as is possible. If you have a conflict with my office hours, please make an appointment. I try to have an open door policy; you are free to come by whenever you choose. If I am busy, I may ask that you make an appointment using email.

IP Policy

To a first approximation, students are the owners of the IP they create. Apart from as required to effectually manage the class logistics (grading, etc), I rescind any personal IP rights I have to students' IP created for this class, if any. I explicitly encourage students to release their projects open source, under a license of their choosing.

Cheating

Academic misconduct of any kind will automatically result in a 0 score on the homework, lab, project, or exam in question and your actions will be reported to

the department chair. Your homework, exams and projects must be individually prepared unless otherwise noted. Posting your assignments to discussion lists is considered academic misconduct. Automated mechanisms are available for checking the originality of source code. Please spend your time trying to solve assigned problems rather than trying to get around the system.

Excuses

Excusing a missed exam or assignment is left to the discretion of the instructor. Illness, family emergencies, and religious observances are examples of acceptable excuses. Computer down time, over sleeping, and social events are examples of unacceptable excuses. Please try to let me know of problems in advance when possible and be prepared to provide verification of your excuse.

Extensions

As a policy, I do not extend due dates of homework and projects. If I choose to do so, I will only announce the extension in class or on piazza. If you hear an extension has been granted and I have not announced it, your information is incorrect.

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 and recursion
- Data Structures
 - Type Systems and Type Checking
 - Composite Types
 - Binding and Scoping
 - Dynamic and Recursive Types

- Basics of JavaScript Programming
 - Call-by-value evaluation
 - Early binding
 - Dynamic types
 - iteration and concurrency
- Basics of Scheme Programming
 - Normal order evaluation
 - Late binding
 - Data driven programming
 - Lists and recursion
 - Higher-order functions
- 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

(Specific topics are subject to change without notice.)