CSE 240 Introduction to Programming Languages
Syllabus and Course Information

Course Coordinator and Instructor

Yinong Chen, Ph.D. (University of Karlsruhe - KIT, Germany)
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University
Tempe, AZ 85287-8809
Office: BYENG M1-06
Phone: (480) 965-2769
Email: yinong@asu.edu, for course related issues, use Canvas mail
URL: http://www.public.asu.edu/~ychen10/

Office and Office Hours

Office: BYENG M1-06 (Brickyard Engineering, Mezzanine floor, entrance from S. 7th street)
Office hours: Monday, 2:30 to 5:15pm
Friday: 9:15 to 11:30am
For appointment and all course related issues, please use Canvas mail, instead of ASU email, so that the email requests can be processed timely.

Catalog Description

Introduces the procedural (C/C++), applicative (Scheme/LISP), and declarative (Prolog) languages. Lecture, lab. Prerequisite: CSE 205.

Objectives and Outcomes

1. Understand features of different programming paradigms
- Students will learn strong vs. weak typing in computer programming languages
- Students will learn control structures of functional, logic, and imperative programming languages.
- Students will learn the execution of functional, logic, and imperative programming languages.
- Students will learn the recursion mechanism of functional, logic, and imperative programming languages.

2. Develop an introductory understanding of an applicative programming language (Scheme)
- Students will work with the Scheme interpreter to evaluate simple functions.
- Students will write and execute simple Scheme functions.
- Students will write and execute Scheme programs requiring multiple functions.

3. Develop an introductory understanding of a declarative programming language (Prolog)

- Students will create a simple Prolog factbase and provide queries to obtain information from the factbase.
- Students will create Prolog programs that use recursive rules to provide a problem solution.
- Students will create Prolog programs that use multiple rules to solve a problem.

4. Develop understanding of procedural and object-oriented programming languages (C/C++)
- Students will write C/C++ programs using pointers.
- Students will write C/C++ programs using multiple functions/procedures.
- Students will write C/C++ programs using dynamic memory allocation.
- Students will write C/C++ programs that allocate and de-allocate static, stack and heap memory.
- Students will design C/C++ programs applying object-oriented features such as inheritance, polymorphism and class hierarchy.

Prerequisite and Requirement

The official prerequisite is CSE205 (Concepts of Computer Science and Data Structures). By topics I expect you to have

- understood basic concepts of computer organization, including registers, memory, arithmetic and logic units, processor, input and output.
- been familiar with object-oriented design, static and dynamic data structures like Integer, Floating-point numbers, Arrays, Strings, Stacks, and data abstraction techniques.
- understood programming techniques and control structures like branching, iteration and recursion.
- commanded a high level programming language like Java or C++ and the environment in which a program is developed, e.g., editor, compiler/interpreter, linker, source code, executable code, debugging tool, etc.

If you do not meet the official prerequisites but are admitted into the course because you did courses "equivalent" to the official prerequisite courses, it is your responsibility to make sure you do understand the necessary background material.

Major Topics Covered in the Course (Tentative)

1. Aspects of programming languages (2 weeks)

1) Different paradigms of programming languages
2) Introduction to the structures of programming languages
3) Program processing: interpretation, compilation and macro processing
4) Typing systems: Strong versus weak typing
5) Orthogonality

2. Introduction to procedural programming languages (C and C++) (4 weeks)

1) Basic data types and data declarations, scope rule and forward declaration
2) Complex data types: array, pointer, string, constants, enumeration, file, and struct types
3) Data structures: stack, linked list, and tree
4) Functions and parameter passing
5) Recursion: concept and programming
6) Time complexity of binary search

3. Introduction to object-oriented programming languages (C++) (3 weeks)

1) Concept of object-oriented programming
2) Class definition and members of class, abstract type, constructor and destructors
3) Memory management: static, stack and heap memory, and garbage collection
4) Inheritance and class hierarchies, polymorphism, virtual functions and dynamic binding
5) Containment versus inheritance
6) Function and operator overloading

4. Introduction to applicative programming language (Scheme) (2.5 weeks)

1) Arithmetic expression and prefix notation
2) Basic Scheme procedures, defining your own procedures
3) Scheme environment, global and local variables, immutable variables and side-effect-free
4) First-class functions: function calls can be placed where its return value is expected
5) Recursive procedures
6) Time complexity of sorting algorithms
7) Programming with data structures, number, character, strings, symbol, pairs and lists
8) High order function, map, reduce, and filter

5. Introduction to declarative programming language Prolog (2.5 weeks)

1) Facts, rules, and goals
2) Structured facts and rules
3) Scope of variables
4) Arithmetic operations
5) Recursion and recursive rules
6) Lists and list manipulation
7) Flow control cut and repeat

Barrett Honors Enrichment Projects

This course offers Barrett Honors Enrichment Projects for students registered with the Barrett Honors college. Students will choose a project in one of these topic areas, for example: C++ multithreading, C# web application development, and Python program development. The same textbook covers these topics. Students will take extra effort to learn and to implement the selected project.

Operating Systems and Programming Language Environments

This course's assignments will require ASU General Unix and Windows operating systems. The programming language environments to be used include GNU GCC and Visual Studio. Students can use their own computers or use ASU lab computers to do the assignments.

Grade Policies

Your performance will be assessed by assignments, programming projects, quizzes, a Mid-Term Exam and a Final Exam. Their weights are:

Assignments and Projects	40%
Weekly Quiz / Exercises	15% (14% + 1%)
Mid-Term	20%
Final Exam	25%

Total	100%

The final letter grade is decided according to the percentage points obtained as follows:

A-, A, A+	90-92, 93-95, 96-100%
B-, B, B+	80-82, 83-85, 86-89%
C, C+	70-75, 76-79%
D	60-69%
E	less than 60%

The grade of "I" (incomplete) can be given ONLY when a student, who is doing otherwise acceptable work (passing grade), is unable to complete a part of work (e.g., the final exam) because of documented illness or other conditions beyond the student's control. In the latter case, the student must discuss with the instructor and complete an application form from the department before the part of work is due or as soon as the circumstances are known. Please see ASU grading policies at: http://students.asu.edu/grades-grading-policies

Extra Credit and Alternative Activity

Missing a graded activity will be given zero credit. In-class exercises and quizzes may not be made up. Extra credits can be given to the entire class, for example, through dropping the two lowest quiz scores and one assignment score. No extra credit-activities will be given to any individual. An alternative to the assignment and exam may be arranged if a student misses the activity and the absence is caused by documented illness or personal emergency that made the completion/attending impossible. A written explanation (including supporting documentation) must be submitted to the instructor before the part of work is due or as soon as the circumstances are known.

Any inquires or appeals on grades of homework, projects, or tests must be done in writing by completing the "Grade Inquiry Form" within a week from the day the assignment was returned or comments were published on-line. State the problem and the rationale for any change in grade in your appeal.

Text, Reference Books and Online Resources

Official textbook: Y. Chen, Introduction to Programming Languages: Programming in C, C++, Scheme, Prolog, C#, and Python, Kendall Hunt Publishing, 6th edition 2019. ISBN 9781524916992 (Hard Copy) and 9781792411762 (eBook)

There are many books and on-line materials that are related to the course. Followings are a few of them that might be useful. Other similar books could be used as reference books too. The first two books were used as the textbooks for this course before the current textbook was approved a few years.

- *Programming Languages Essentials*, H. Bal and D. Grune, Addison-Wesley, 1994. This book focuses on the language principles. Little programming is taught.
- *MiniManuals in PC SCHEME* and PROLOG, McGraw-Hill, 1991.
- The C Programming Language, B. Kernighan and D. Ritchie, Prentice-Hall, 1978.
- C++ Programming with Design Patterns Revealed, T. Mueldner, Addison Wesley, 2002.
- *C++: How to Program*, Third Edition, Harvey and Paul Deitel, Prentice-Hall, 2001.
- *The C++ Programming Language*, B. Stroustrup, Addison-Wesley, 1997.
- *The Schematics of Computation*, V. Manis and J. Little, Prentice Hall, 1995.

- *Programming in Prolog*, W.F. Clocksin, and C.S. Mellish, Third, revised and extended edition, Springer-Verlag
- The C++ Resources Network Good tutorial on basics of C++. Also has solid documentation for the Standard Template Library (STL), http://www.cplusplus.com/doc/tutorial/
- Stack Overflow Good site for finding answers to specific problems, http://stackoverflow.com/

Academic Integrity and Honor Code

You are encouraged to cooperate in study group on learning the course materials. However, you may not cooperate on preparing the individual assignments. Anything that you turn in must be your own work: You must write up your own solution with your own understanding. If you use an idea that is found in a book or from other sources, or that was developed by someone else or jointly with some group, make sure you acknowledge the source and/or the names of the persons in the write-up for each problem. When you help your peers, you should never show your work to them. All assignment questions must be asked in the course discussion board. Asking assignment questions or making your assignment available in the public websites before the assignment due will be considered cheating. All individual tests must be done independently. Working together during tests is not permitted.

The instructor and the TA will **CAREFULLY** check any possible proliferation or plagiarism by comparing among the student submissions, previous student submissions, and the publications in the public Web sites. We will use the document/program comparison tools like MOSS (Measure Of Software Similarity: http://moss.stanford.edu/) to check all assignments and tests that you submitted for grading.

Fulton Schools of Engineering Honor Code (http://engineering.asu.edu/integrity/honor-code/)

1. Seek out, acquaint myself with, and obey the instructor's rules concerning the materials I am allowed to use and the types of collaboration in which I am permitted to engage in each of my courses.
2. Help my fellow engineering students to succeed both academically and professionally, while both following the instructor's guidelines on collaboration and encouraging my classmates to behave ethically.
3. Ensure that all of my individual work products reflect my own abilities and not those of someone else. I will never copy the work of others or give others the opportunity to copy mine.
4. Contribute a fair share of work to all teamwork in which I participate, and acknowledge the contributions of others. I will accept responsibility for the integrity of all work submitted by my team.
5. Use only aids authorized by the instructor during all examinations, quizzes, projects, assignments and other evaluations.
6. Provide aid to, or receive aid from other students only as permitted by the instructor.
7. Give full credit to others for their words and ideas, whether directly quoted or paraphrased, using proper citation practices in all of my work, including text, figures and computer code, and all materials obtained from the Internet.

8. Never act dishonestly including lying, cheating, stealing, or attempting to corrupt the academic enterprise in any way.
9. Ensure that all data I record or report are objective, true, accurate and properly documented.
10. Treat all students, faculty and staff with respect, courtesy and dignity, the way I would like to be treated myself.
11. Recognize that it is how I act when no one else is watching that defines my true character.
12. Act at all times with integrity, as the true professional that I am to become.

Policy against threatening behavior, per the Student Services Manual, SSM 104–02

Students, faculty, staff, and other individuals do not have an unqualified right of access to university grounds, property, or services. Interfering with the peaceful conduct of university-related business or activities or remaining on campus grounds after a request to leave may be considered a crime. All incidents and allegations of violent or threatening conduct by an ASU student (whether on- or off-campus) must be reported to the ASU Police Department (ASU PD) and the Office of the Dean of Students.

Disability Accommodations.

Suitable accommodations will be made for students having disabilities. Students needing accommodations must register with the ASU disabilities resource Center and provide documentation of that registration to the instructor. Students should communicate the need for an accommodation in sufficient time for it to be properly arranged.

Harassment and Sexual Discrimination

Arizona State University is committed to providing an environment free of discrimination, harassment, or retaliation for the entire university community, including all students, faculty members, staff employees, and guests. ASU expressly prohibits discrimination, harassment, and retaliation by employees, students, contractors, or agents of the university based on any protected status: race, color, religion, sex, national origin, age, disability, veteran status, sexual orientation, gender identity, and genetic information.

Title IX is a federal law that provides that no person be excluded on the basis of sex from participation in, be denied benefits of, or be subjected to discrimination under any education program or activity. Both Title IX and university policy make clear that sexual violence and harassment based on sex is prohibited. An individual who believes they have been subjected to sexual violence or harassed on the basis of sex can seek support, including counseling and academic support, from the university. If you or someone you know has been harassed on the basis of sex or sexually assaulted, you can find information and resources at https://sexualviolenceprevention.asu.edu/faqs.

Mandated sexual harassment reporter: As an employee of the University I am considered a mandated reporter and therefore obligated to report any information regarding alleged acts of sexual discrimination that I am informed of or have a reasonable basis to believe occurred.

ASU Counseling Services, https://eoss.asu.edu/counseling, is available if you wish to discuss any concerns confidentially and privately.