CSE 240
Introduction to Programming Languages
Spring 2018

Javier Gonzalez-Sanchez, Ph.D.

Office:	Peralta Hall Room 230U
Office hours:	By appointment
E-mail:	javiergs@asu.edu
Web:	javiergs.engineering.asu.edu

1. Description

This course introduces diverse paradigms of programming languages, including procedural (Language C), object-oriented (Language C++), functional (Language LISP/Scheme), and logic (Language Prolog) paradigm.

2. Credits and contact hours

3 credits, lecture/lab

3. Prerequisites or co-requisites

CSE 205 – Object-Oriented Programming and Data Structures

Topics that you are expect to have learned before taking this class includes:

- 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.
- proficient in 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 don't 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.

4. Topics

1. Introduction to programming languages
   - Concepts, features and paradigms
   - History of programming languages
   - Structure of programming languages: lexical, syntactic, and semantic
   - Code generation and interpreters
   - Data types and type checking (strong versus weak typing).

2. Procedural programming languages (Language C)
   - Procedural programming
   - Control structures
   - Primitive data types and data declarations
   - Complex data types: array, pointer, string, constants, enumeration, struct types
   - Compound data structures: stack, linked list, and tree
   - Files
   - Functions and parameter passing
   - Recursion
   - Modular design

3. Object-oriented programming languages (Language C++)
   - Object-oriented programming
   - Class definition and members of class, abstract type, constructor and destructors
   - Memory management: static, stack and heap memory, and garbage collection
   - Inheritance and class hierarchies, polymorphism, virtual functions and dynamic binding
   - Containment versus inheritance
   - Function and operator overloading
   - File operators
   - Exception handling

4. Functional programming languages (Language LISP/Scheme)
   - Functional programming
   - Prefix notation
   - Arithmetic expression
   - Procedures
   - Global and local variables, immutable variables and side-effect-free
   - Data structures, number, character, strings, symbol, pairs and lists
   - First-class functions
   - Recursive procedures
   - High order functions

5. Logic programming languages (Language Prolog)
   - Introduction to logic programming
   - Facts, rules, and goals

- Arithmetic operations
- Recursion and recursive rules
- Lists and list manipulation
- Flow control

5. Learning 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 (Lisp/Scheme)
   - Students will work with the Lisp/Scheme interpreter to evaluate functions.
   - Students will write and execute Lisp/Scheme functions.
   - Students will write and execute Lisp/Scheme programs requiring multiple functions.

3. Develop an introductory understanding of a declarative programming language (Prolog)
   - Students will create a Prolog fact-base and provide queries to obtain information from the fact-base.
   - 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 that allocate and de-allocate static, stack and heap
   - Students will design C/C++ programs applying object-oriented features such as inheritance, polymorphism and class hierarchy.

6. Course resources

- Textbook: **Introduction to Programming Languages: Programming in C, C++, Scheme, Prolog, C#, and SOA**, Y. Chen, W.T. Tsai, Kendall Hunt Publishing, 5th edition. 2017.

- **Programming Languages Essentials**, H. Bal and D. Grune, Addison-Wesley, 1994.

- **The C Programming Language**, B. Kernighan and D. Ritchie, Prentice-Hall, 1978.

- **The C++ Programming Language**, B. Stroustrup, Addison-Wesley, 1997.

- **C++: How to Program**, Third Edition, H. Deitel and P. Deitel, Prentice-Hall, 2001.

- **Programming in Prolog**, W. Clocksin and C. Mellish, Third edition, Springer, 2003

- **ANSI Common Lisp,** P. Graham, First Edition, Prentice Hall, 1995.

7. Evaluation

Your course grade in CSE240 will be based on the distribution below.

Exams	50%
Quizzes	25%
Homework (Programming Projects)	25%

Exams: There will be **two exams** (a midterm and a final exam) given during the semester. The exams will be comprehensive. Exams may consist of multiple choice, fill-in-the-blank, short answer, programming problems, or any combination thereof.

Assignments and projects: They will be given throughout the semester. They are designed to help you practice the concepts you need to learn in this class. Finding the answer somewhere does not equal practice. Programming projects should take you anywhere from thirty minutes to ten hours to complete, depending on your skill level and study habits. They may include a written portion, a programming portion, or both. If you have trouble, please take advantages of office hours and other help resources, such as the discussion board. Assignments and projects **can never be made up** and **can never be turned in late**.

Quizzes: Quizzes may be announced during the class or they may be **unannounced**. There are **no make-ups** for missed quizzes.

8. Grading

The following is the grading scale that will be used in the course (I reserve the right to modify the scale or grading system). Any adjustments to grading must be presented within two days of receipt of your graded materials – no grade changes will be made afterwards.

- A+ >=   97%
- A   >=   93%
- A- >=   89%
- B+ >=   85%
- B   >=   81%
- B- >=   77%
- C+ >=   73%

- C >= 69%
- D >= 65%
- E < 65%

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: https://students.asu.edu/grades

Grading Appeals: Any discrepancy or disagreement in grading must be presented to the instructor within a week of your receipt of your graded materials; otherwise no grade change will be made.

Note: I reserve the right to change this grading system as the course progresses and various circumstances develop.

9. Classroom Behavior

Cell phones and pagers (must be/or state alternative rule) turned off during class to avoid causing distractions. **The use of recording devices is not permitted during class**. Any violent or threatening conduct by an ASU student in this class will be reported to the ASU Police Department and the Office of the Dean of Students

10. Disability Accommodations.

Suitable accommodations will be made for students having disabilities and students should notify the instructor as early as possible if they will require same. Such students must be registered with the Disability Resource Center and provide documentation to that effect.

11. 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.

The Ira A. Fulton Schools of Engineering expect all students to adhere to ASU's policy on Academic Dishonesty. These policies can be found in the Code of Student Conduct:

https://provost.asu.edu/academic-integrity

ALL cases of cheating or plagiarism will be handed to the Dean's office. Penalties include a failing grade in the class, a note on your official transcript that shows you were punished for cheating, suspension, expulsion and revocation of already awarded degrees.

Fulton Schools of Engineering Honor Code

https://engineering.asu.edu/integrity/

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.

12. Sexual Discrimination

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

As a mandated reporter, I am obligated to report any information I become aware of regarding alleged acts of sexual discrimination, including sexual violence and dating violence. ASU Counseling Services, https://eoss.asu.edu/counseling, is available if you wish discuss any concerns confidentially and privately.

IMPORTANT

- Any information in this syllabus (other than grading and absence policies) may be subject to change with reasonable advance notice.

- All contents of these lectures, including written materials distributed to the class, are under copyright protection. Notes based on these materials may not be sold or commercialized without the express permission of the instructor. [ACD 304-06].