

COMP 4040 – 6040: Programming Languages

Contact Information

Instructor: Qishi Wu
Office/Lab: 374/206 Dunn Hall
E-mail: gishiwu@memphis.edu
Phone: 678-1539

Department office: 209 Dunn Hall
Department phone: 678-5465

Course website: <http://www.cs.memphis.edu/~gishiwu>

Course Description

Comparative features, syntax, and applicability of high-level programming languages (C/C++, JAVA, C#, JavaScript, Fortran, Pascal, PHP, Perl, Lisp, Prolog, etc.), programming preliminaries, variables, data types, expressions, control structures, subprograms, runtime environment, dataflow, recursion, object-oriented programming, concurrency, exception and event handling, functional and logic programming.

This course helps the students gain a deep understanding of the main concepts in a wide variety of programming languages and improves the students' programming skill by showing the inner behavior of a computing system during the execution of programs.

Prerequisite

COMP 3160 or permission of instructor.

Required Textbook

Concepts of Programming Languages (8th Edition), Robert W. Sebesta, Addison-Wesley.

Other Resources

Additional reading materials including books, journal/conference papers, and websites will be assigned for some advanced topics as the course proceeds.

Evaluation

Grading components:

Attendance	10%
Homework	10%
Project	20% [W]
Midterm	30%
Final	30%

[W]: Students are required to turn in an essay, discussing concepts of programming languages. Specific topics are to be announced after the mid-term exam.

Grading scale:

Grade	COMP 4040	COMP 6040
A	85 – 100	90 – 100
B	75 – 84	80 – 89
C	65 – 74	70 – 79
D	55 – 64	60 – 69
F	54 and below	59 and below

Final grades will not be curved unless necessary.

Late Policy:

Students are expected to complete work on schedule. Late work is not accepted unless prior arrangements are made with the instructor.

Plagiarism/Cheating Policy

Plagiarism or cheating behavior in any form is unethical and detrimental to proper education and **will not be tolerated**. All work submitted by a student (projects, programming assignments, lab assignments, quizzes, tests, etc.) is expected to be a student's own work. The plagiarism is incurred when any part of anybody else's work is passed as your own (no proper credit is listed to the sources in your own work) so the reader is led to believe it is therefore your own effort. Students are allowed and encouraged to discuss with each other and look up resources in the literature (including the Internet) on their assignments, but **appropriate references must be included for the materials consulted**, and appropriate citations made when the material is taken verbatim.

If plagiarism or cheating occurs, the student will receive a failing grade on the assignment and (at the instructor's discretion) a failing grade in the course. The course instructor may also decide to forward the incident to the University Judicial Affairs Office for further disciplinary action. For further information on U of M code of student conduct and academic discipline procedures, please refer to: <http://www.people.memphis.edu/~jaffairs>.

Course Syllabus

Week	Chapter	Topic
1	1, 2	<ul style="list-style-type: none">• Introduction, programming preliminaries
2	5, 6	<ul style="list-style-type: none">• Variables, storage binding, scope, type• Data types: primitive, user-defined, string, array, record, union, pointer, reference
3, 4	7, 8	<ul style="list-style-type: none">• Expressions, operators and operands, precedence, associativity, relational and Boolean expression, short-circuit evaluation, assignment statements• Statement-level control structures: selection, iterative, unconditional
5, 6, 7	9, 10	<ul style="list-style-type: none">• Subprograms: introduction, fundamentals, design issues, local variables, parameter-passing methods, overloaded operations• Implementation of subprograms: runtime stack, activation record, nested subprograms, blocks, dynamic scoping
8		<ul style="list-style-type: none">• Review and midterm exam
9, 10	11, 12	<ul style="list-style-type: none">• Abstract data types, encapsulation constructs• Object-oriented programming
11	13, 14	<ul style="list-style-type: none">• Concurrency• Exception and event handling
12	3, 4	<ul style="list-style-type: none">• Syntax and semantics• Lexical analysis, parsing problem
13	15	<ul style="list-style-type: none">• Functional programming languages
14	16	<ul style="list-style-type: none">• Logic programming languages
15		<ul style="list-style-type: none">• Review