Course Description:
COMP 4040/6040 Comparative features, syntax and applicability of high-level programming languages such as C, C++, JAVA, PHP, JavaScript, and study of data types, data structures, procedures, recursion, runtime environment, string manipulation, list processing, array processing, documentation, programming style.

PREREQUISITE: COMP 3160

Why This Course?
Programming languages are the medium through which programmers precisely describe concepts, formulate algorithms, and reason about solutions. In the course of a career, a computer scientist will work with many different languages, separately or together. Software developers must understand the programming models underlying different languages and make informed design choices in languages supporting multiple complementary approaches. Computer scientists will often need to learn new languages and programming constructs, and must understand the principles underlying how programming language features are defined, composed, and implemented. The effective use of programming languages, and appreciation of their limitations, also requires a basic knowledge of programming language translation and static program analysis, as well as run-time components such as memory management.

Learning Outcomes
• Ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs/requirements.
• An ability to use current techniques, skills, and tools necessary for computing practice.
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.

Resources:
Required textbooks
CONCEPTS OF PROGRAMMING LANGUAGES 10/E ROBERT W. SEBESTA
**Evaluation:**
Pre-midterm exam: 17.5%
Midterm: 27.5%,
Pre-final exam: 17.5%
Final Exam: 37.5%

Grading Scale (% of the best student score):
A: 90 and above, B: 80 to 89, C: 70 to 79, D: 60 to 69, F: < 60.

**Course Topics:**

Week 1: Overview of chapters 1-5

Week 2: Chapter 6 Data Types

Week 3: Chapter 7 Expressions and Assignment Statements

Week 4: Chapter 8 Statement-Level Control Structures

Week 5: Chapter 9 Subprograms

Week 6: Chapter 10 Implementing Subprograms

Week 7: Chapter 11 Abstract Data Types and Encapsulation Constructs

Week 8 and 9: Chapter 12 Support for Object-Oriented Programming

Week 10 and 11: Chapter 13 Concurrency

Week 12: Chapter 14 Exception Handling and Event Handling

Week 13: Chapter 15 Functional Programming Languages

Week 14: Chapter 16 Logic Programming Languages

**Course Policies:**
Cell-phones are not to be utilized during class. This includes making or receiving calls and text-messaging.

**Testing Policy**
There will NOT be any makeup exams unless there is a documented emergency, so it is very important for you to attend every lecture and exam.

**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 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.
For further information on U of M code of student conduct and academic discipline procedures, please refer to:
http://libguides.memphis.edu/academicintegrity

Disability Notice:
Any student who anticipates physical or academic barriers based on the impact of a disability is encouraged to speak with me privately. Students with disabilities should also contact Disability Resources for Students (DRS) at 110 Wilder Tower (901-678-2880).