CMPS 401
Survey of Programming Languages
Fall 2008

Instructor: Dr. Kuo-pao Yang
Office: 327C Fayard Hall
Telephone: (985) 549 - 5088
Email: kyang@selu.edu
Web Page: http://www2.selu.edu/Academics/Faculty/kyang/
Office Hours: T, TH 12:15PM - 03:30PM, 06:15PM - 06:45PM


Course Description: Prerequisite: Computer Science 390. Involves the formal study of programming languages, specification, and analysis in terms of data types and structures.

Minimum Topics:
• History of programming languages
• Brief survey of programming paradigms
  o Procedural languages
  o Object-oriented languages
  o Functional languages
  o Declarative, non-algorithmic languages
  o Scripting languages
• The effects of scale on programming methodology
• The concept of a virtual machine
• Intermediate languages
• Comparison of interpreters and compilers
• Machine-dependent and machine-independent aspects of translation
• Introduction to syntax and semantics
• The conception of types as a set of values with together with a set of operations
• Declaration models (binding, visibility, scope, and lifetime)
• Overview of type-checking
• Procedures, functions, and iterators as abstraction mechanisms
• Parameterization mechanisms (reference vs. value)
• Type parameters and parameterized types
• Data types
• Expressions and assignment statements
• Control statements (selection and iterative)
• Subprograms in programming languages

Learning Objectives: Students will be able to:
• Understand the evolution of programming languages and relate how this history has led to the paradigms available today.
• Identify at least one outstanding and distinguishing characteristic for each of the programming paradigms covered in this unit.
• Evaluate the tradeoffs between the different paradigms, considering such issues as space efficiency, time efficiency (of both the computer and the programmer), safety, and power of expression.
• Describe the importance and power of abstraction in the context of virtual machines.
• Explain the benefits of intermediate languages in the compilation process.
• Evaluate the tradeoffs in performance vs. portability.
• Compare and contrast compiled and interpreted execution models, outlining the relative merits of each.
• Describe the phases of program translation from source code to executable code and the files produced by these phases.
• Explain the differences between machine-dependent and machine-independent translation and where these differences are evident in the translation process.
• Explain formal methods of describing syntax (backus-naur form, context-free grammars, and parser tree).
• Describe the meanings of programs (dynamic semantics, weakest precondition).
• Identify and describe the properties of a variable such as its associated address, value, scope, persistence, and size.
• Discuss type incompatibility.
• Explain data types: primitive types, character string types, user-defined ordinal types, array types, associative arrays, record type, union types, point and reference types
• Demonstrate different forms of binding, visibility, scoping, and lifetime management.
• Defend the importance of types and type-checking in providing abstraction and safety.
• Explain how abstraction mechanisms support the creation of reusable software components.
• Demonstrate the difference between overridden and overloaded subprograms
• Explain functional side effects.
• Demonstrate the difference between pass-by-value, pass-by-result, pass-by-value-result, pass-by-reference, and pass-by-name parameter passing.
• Explain the difference between the static binding and dynamic binding.

Course Learning Outcomes and Evaluation:

• An ability to apply knowledge of computing and mathematics appropriate to the discipline
  o Justification: Assignments require programming in various languages. Mathematical analysis and metrics also exist in this course.
• An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
  o Justification: Students learn how to evaluate advantages and disadvantages of different languages in order to select the most appropriate language.
• An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs
  o Justification: This is largely the intent of the homework/programming/project assignments in this course.
• Recognition of the need for, and an ability to engage in, continuing professional development
  o Justification: Some assignments are purposely designed to require research to obtain certain functions of programming languages that can lead to the solutions sought. Also, emphasis is placed on fundamentals of languages to support future ability to follow developments.
• An ability to use current techniques, skills, and tools necessary for computing practice
  o Justification: Current techniques, computing environments, and programming languages are used throughout the assignments in this course.
Schedule (Tentative):

<table>
<thead>
<tr>
<th>Week #</th>
<th>Week of</th>
<th>Topics to be Covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>08/20</td>
<td>Chap01: Preliminaries</td>
</tr>
<tr>
<td>2</td>
<td>08/25</td>
<td>Chap01: Preliminaries</td>
</tr>
<tr>
<td>3</td>
<td>09/01</td>
<td>Chap02: Evolution of the Major Programming Languages</td>
</tr>
<tr>
<td>4</td>
<td>09/08</td>
<td>Chap03: Describing Syntax and Semantics</td>
</tr>
<tr>
<td>5</td>
<td>09/15</td>
<td>Chap04: Lexical and Syntax Analysis</td>
</tr>
<tr>
<td>6</td>
<td>09/22</td>
<td>Chap05: Names, Bindings, Type Checking, and Scopes</td>
</tr>
<tr>
<td>7</td>
<td>09/29</td>
<td>Midterm Exam; 10/2-10/3 Fall Break</td>
</tr>
<tr>
<td>8</td>
<td>10/06</td>
<td>Chap06: Data Types</td>
</tr>
<tr>
<td>9</td>
<td>10/13</td>
<td>Chap07: Expressions and Assignment Statements</td>
</tr>
<tr>
<td>10</td>
<td>10/20</td>
<td>Project Presentation; Chap07: Expressions and Assignment Statements</td>
</tr>
<tr>
<td>11</td>
<td>10/27</td>
<td>Project Presentation; Chap08: Statement-Level Control Structures</td>
</tr>
<tr>
<td>12</td>
<td>11/03</td>
<td>Chap08: Statement-Level Control Structures</td>
</tr>
<tr>
<td>13</td>
<td>11/10</td>
<td>Chap09: Subprograms</td>
</tr>
<tr>
<td>14</td>
<td>11/17</td>
<td>Chap09: Subprograms</td>
</tr>
<tr>
<td>15</td>
<td>11/24</td>
<td>Chap10: Implementing Subprograms; 11/26-12/28 Thanksgiving Holiday</td>
</tr>
<tr>
<td>16</td>
<td>12/01</td>
<td>Chap10: Implementing Subprograms</td>
</tr>
<tr>
<td>17</td>
<td>12/08</td>
<td>Final Exams</td>
</tr>
</tbody>
</table>

10/24/2008: Last day to withdraw or resign from the University
12/15/2008: Last day to return rental textbooks without a fine

Grading Policy:

<table>
<thead>
<tr>
<th>Component</th>
<th>Weightage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programs/Homework/Quizzes</td>
<td>40%</td>
</tr>
<tr>
<td>Project</td>
<td>20%</td>
</tr>
<tr>
<td>Midterm</td>
<td>20%</td>
</tr>
<tr>
<td>Final Examinations</td>
<td>20%</td>
</tr>
</tbody>
</table>

A: 90 – 100 %; B: 80 – 89 %; C: 70 – 79 %; D: 60 – 69 %; F: 0 – 59 %

Attendance Policy: Attendance is mandatory for all sessions of this course. Absences will be excused only with a valid written excuse, such as from a physician. [http://www2.selu.edu/Academics/Depts/FacSen/attendencepolicy.pdf](http://www2.selu.edu/Academics/Depts/FacSen/attendencepolicy.pdf)

Program/Homework: Programs/Homework will be collected periodically. Selected problems will be graded. No late Programs / homework will be accepted.

Quizzes: There will be a number of quizzes during this term. No makeup quizzes will be given.

Examinations: There will be a midterm and a final examination. No makeup examinations will be given. If you miss an examination with an excused absence, your grade for this missed examination will be replaced of your final examination. In the case of illness, doctor’s excuse is needed. If your absence is unexcused, you will receive a grade of zero on the examination you missed.

Academic Dishonesty: Cases involving alleged academic dishonesty will be dealt with according to established university policies. Classroom behavior that is determined inappropriate and cannot be resolved by the student and the faculty member may be referred for administrative or disciplinary review. Cheating on examinations, plagiarism, improper acknowledgment of sources in essays and the use of a single essay or paper in more than one course without permission are considered very serious offenses and shall be grounds for disciplinary action. Additional information about the Code of Student Conduct may be found at: [http://www.selu.edu/admin/jud_affairs/assets/code_conduct.pdf](http://www.selu.edu/admin/jud_affairs/assets/code_conduct.pdf)

ADA Accommodation: If you are a qualified student with a disability seeking accommodations under the Americans with Disabilities Act, you are required to self-identify with the Office of...
Disability Services, Room 203, Student Union. No accommodations will be granted without documentation from the Office of Disability Services.

Classroom Decorum: Free discussion, inquiry, and expression are encouraged in this class. Classroom behavior that interferes with either (a) the instructor’s ability to conduct the class or (b) the ability of students to benefit from the instructor is not acceptable. Examples include routinely entering class late or departing early; use of beepers, cellular telephones, or other electronic devices; repeatedly talking in class without being recognized; talking while others are speaking; or arguing in a way that is perceived as “crossing the civility line.” In the event of a situation where a student legitimately needs to carry a beeper or cellular telephone to class, prior notice and approval of the instructor is required.

The office/classroom is not a place for children and neither employees nor students are to bring their family members for day care or baby sitting. If children require care, then the employee/student is expected to provide that care in an environment other than Southeastern office/classroom space.