Organization of Programming Languages  
CSC 408  
Fall 2016  
**Time:** 06:30 PM-09:15 PM T, **Location:** LH 103, **Credit:** 3 units

**Contacting Your Instructor(s)**  
**Instructor:** Dr. Bo Li  
**Office:** Science Building, Room 105D  
**Office hours:** 2:00-6:00 PM T&W, 3:00-5:00 PM R, or by appointment  
**Email:** bo.li@usm.edu  
**Phone:** 228-214-3306

**Course coordinator:** Mr. Tom Rishel  
Email Address: Tom.rishel@usm.edu

**Catalog Description**  
**Prerequisite(s):** [CSC 307](#). Runtime behavior of programs, formal specification and comparative analysis of programming languages, programming paradigms.

CSC 307 - Data Structures and Algorithm Analysis  

**Course Description and Objectives**  
3 hrs. Runtime behavior of programs, formal specification and comparative analysis of programming languages, programming paradigms. Programming Intensive Course. At least 4 programs will be assigned.

This course is a survey of the fundamental concepts underlying modern programming languages. The emphasis is on identifying common syntactic and semantic language constructs and features, examining their manifestation in specific programming languages, and understanding the similarities and differences that arise in language implementations. The programming paradigms that will be studied include:

- Procedural / imperative
- Functional
- Object-oriented
- Concurrent

The representative language features will be drawn from ANSI C, C++, Java, Scheme, ML, and JavaScript.

**Course Materials**  
Course Outcomes

- Summarize the evolution of programming languages illustrating how this history has led to the paradigms available today.
- Identify at least one 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.
- Distinguish between programming-in-the-small and programming-in-the-large.
- Compare and contrast compiled and interpreted execution models, outlining the relative merits of each.
- Describe the phases of program translation form 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 the value of declaration models, especially with respect to programming-in-the-large.
- Identify and describe the properties of a variable such as its associated address, value, scope, persistence, and size.
- Discuss type incompatibility.
- Demonstrate different forms of binding, visibility, scooping, and lifetime management.
- Defend the importance of types and type checking in providing abstraction and safety.
- Evaluate tradeoffs in lifetime management (reference counting vs. garbage collection).
- Explain how abstraction mechanisms support the creation of reusable software components.
- Demonstrate the difference between call-by-value and call-by-reference parameter passing.
- Defend the importance of abstractions, especially with respect to programming-in-the-large.
- Describe how the computer system uses activation records to manage program modules and their data.
- Justify the philosophy of object-oriented design and the concepts of encapsulation, abstraction, inheritance, and polymorphism.
- Compare and contrast the notions of overloading and overriding methods in an object-oriented language.
Outline the strengths and weaknesses of the functional programming paradigm.

Explain the use of functions as data, including the concept of closures.

Design, code, test, and debug programs using the functional paradigm.

**Course Workload Statement**

Students are expected to invest considerable time outside of class in learning the material for this course. The expectation of the University of Southern Mississippi is that each week students should spend approximately 2-3 hours outside of class for every hour in class working on reading, assignments, studying, and other work for the course. We realize that most students work and have family or other obligations. Time management is thus critical for student success. All students should assess their personal circumstances and talk with their advisors about the appropriate number of credit hours to take each term, keeping in mind that 30 credit hours each year are needed to graduate in four years. Resources for academic support can be found at [https://www.usm.edu/success](https://www.usm.edu/success).

**Grading Policies and Calculation**

A list of possible grades at the University can be found in the Bulletin ([http://catalog.usm.edu](http://catalog.usm.edu)). Note that students will receive an “interim grade” at the six-week point to give them an indication of their performance at that point in the semester.

Students may drop a course with no penalty in the first week of the semester. If students wish to leave a course with a grade of “W” (for “withdrawal”), they may request to do so before the 50th day (specific dates can be found here: [https://www.usm.edu/registrar/calendars](https://www.usm.edu/registrar/calendars)).

Important note: Students who receive a grade of W do not receive any money back and that grade is permanently included on their transcripts.

Students should be aware that “Incompletes” can only be assigned in cases of “extraordinary circumstances” beyond the student’s control.

<table>
<thead>
<tr>
<th>Items</th>
<th>Percentage</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>40%</td>
<td>See class schedule</td>
</tr>
<tr>
<td>Mid-term Exam</td>
<td>30%</td>
<td>Oct 4, 2016</td>
</tr>
<tr>
<td>Final Exam</td>
<td>30%</td>
<td>December 6, 2016</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A grade</td>
<td>&gt;= 90%</td>
</tr>
<tr>
<td>B grade</td>
<td>80%-89%</td>
</tr>
<tr>
<td>C grade</td>
<td>70%-79%</td>
</tr>
<tr>
<td>D grade</td>
<td>60-69%</td>
</tr>
<tr>
<td>F grade</td>
<td>&lt; 60%</td>
</tr>
</tbody>
</table>
Grading Policy for Programming Assignments (up to the number of points given, dependent upon quality)

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complies and Links without error</td>
<td>10</td>
</tr>
<tr>
<td>Executes without error</td>
<td>20</td>
</tr>
<tr>
<td>Performs correctly</td>
<td>20</td>
</tr>
<tr>
<td>Appropriate commenting</td>
<td>10</td>
</tr>
<tr>
<td>Following implementation instructions</td>
<td>20</td>
</tr>
<tr>
<td>Logically organized and formatted</td>
<td>10</td>
</tr>
<tr>
<td>Turned on time</td>
<td>10</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100 points</strong></td>
</tr>
</tbody>
</table>

**Academic Integrity Statement**

All students at the University of Southern Mississippi are expected to demonstrate the highest levels of academic integrity in all that they do. Forms of academic dishonesty include (but are not limited to):

- Cheating (including copying from others’ work)
- Plagiarism (representing another person’s words or ideas as your own; failure to properly cite the source of your information, argument, or concepts)
- Falsification of documents
- Disclosure of test or other assignment content to another student
- Submission of the same paper or other assignment to more than one class without the explicit approval of all faculty members’ involved
- Unauthorized academic collaboration with others
- Conspiracy to engage in academic misconduct

Engaging in any of these behaviors or supporting others who do so will result in academic penalties and/or other sanctions. If a faculty member determines that a student has violated our Academic Integrity Policy, sanctions ranging from resubmission of work to course failure may occur, including the possibility of receiving a grade of “XF” for the course, which will be on the student’s transcript with the notation “Failure due to academic misconduct.” For more details, please see the University’s Academic Integrity Policy: [https://www.usm.edu/institutional-policies/policy-acaf-pro-012](https://www.usm.edu/institutional-policies/policy-acaf-pro-012)

Note that repeated acts of academic misconduct will lead to expulsion from the University.

**Academic Support Resources**

Please see our Student Success Website: [http://www.usm.edu/success](http://www.usm.edu/success) for information on where you can find tutoring and other academic assistance, as well as the location of key resources on campus.

If a student has a disability that qualifies under the Americans with Disabilities Act (ADA) and requires accommodations, he/she should contact the Office for Disability Accommodations (ODA) for information on appropriate policies and
procedures. Disabilities covered by ADA may include learning, psychiatric, physical disabilities, or chronic health disorders. Students can contact ODA if they are not certain whether a medical condition/disability qualifies.

**Address:**
The University of Southern Mississippi
Office for Disability Accommodations
118 College Drive # 8586
Hattiesburg, MS 39406-0001
Gulf Coast: 228-214-3232
Voice Telephone: 601.266.5024 or 228.214.3232 Fax: 601.266.6035
Individuals with hearing impairments can contact ODA using the Mississippi Relay Service at 1.800.582.2233 (TTY) or email ODA at oda@usm.edu.

**Important Class Policies**

**Class Attendance**
Attendance will be taken every class period. Students who are excessively absent (30% of all the lectures, not including exams) and/or tardy will be assigned a grade of NA (Not Attending) according to the University’s Class Attendance Policy. Coming to class unduly late and leaving class unduly early is treated the same as being absent.

**Plagiarism**
Plagiarism or cheating of any type will not be tolerated. This includes, but is not limited to, copying programs, projects, assignments, abstracts, documentation, wandering eyes/copying on tests, turning in previously submitted term papers or projects (in whole or part), using other person's USM computer accounts to do projects, programs, etc., getting other people to do your assignments, etc. Copying from the internet of any type is not allowed.

**Missing Exams**
If you should miss an exam, you must let me know why you will be missing the exam before the exam is administered to the class (at least a couple of days before). You may send me an e-mail, or come by my office to explain why you will be missing the exam. If the excuse is reasonable, (I am the sole judge of reasonability) I will allow you to retake the exam. Please be aware that I am not responsible for lost e-mail. It is your responsibility to make sure that I know you are missing the exam.

Once I agree upon your excuse, please furnish me with a confirmation of your reason(s) for missing the exam. This confirmation must be in my hands by a maximum of 5 days after the exam is administered.

**Tardiness for Tests**
It is your responsibility to make it to class on time for all scheduled examinations. If you are late for an examination, you will be allowed only the remainder of the scheduled period to complete the examination.
Cell Phones/Beepers/Pagers/etc.
Please make sure you switch off all cell phones/beepers/pagers while you are in class. I may ask you to leave the class under such circumstances.

Turning In Work
Assignments not turned on time will not receive full credit. All assignments are submitted via Blackboard. E-mail submission is not accepted. All homework assignments are due the end of the due dates (please refer to the class schedule for more details. Note: some due dates may be adjusted, then please follow related Blackboard announcements).

Test Regrading Policy
I will be glad to review any test for possible grading errors. Any requests for regrading of tests must be made within one calendar week upon the return of the test to the class, regardless of when you received your test back. If you submit your test for regrading, I reserve the right to regrade your entire test.

More Than 2 Finals on the Same Day
If you have two or more final examinations scheduled for the same day and you wish to reschedule my examination, please inform me.

E-mail Addresses
I may contact you during the whole semester and will use the email address provided on SOAR.

Mental Well-Being Statement
USM recognizes that students sometimes experience challenges that make learning difficult. If you find that life stressors such as anxiety, depression, relationship problems, difficulty concentrating, alcohol/drug problems, or other stressful experiences are interfering with your academic or personal success, consider contacting Student Counseling Services on campus at 601-266-4829. More information is also available at https://www.usm.edu/student-counseling-services. All students are eligible for free, confidential individual or group counseling services. *In the event of emergency, please call 911 or contact the counselor on call at 601-606-HELP (4357).*

Class Schedule

Tentative syllabus

General Topics to Cover* (subject to restrictions and modifications):
- Overview of programming languages
  - History of programming languages
  - Brief survey of programming paradigms

* The topics are taken selectively from ACM Computing Curricula 2001 as direct quotes.
- Procedural languages
- Object-oriented languages
- Functional languages
- Declarative, non-algorithmic languages
- Scripting languages
  - The effects of scale on programming methodology
- Introduction to language translation
  - Comparison of interpreters and compilers
  - Language translation phases (lexical analysis, parsing, code generation, optimization)
  - Machine-dependent and machine-independent aspects of translation
- Declarations and types
  - The conception of types as a set of values together with a set of operations
  - Declaration models (binding, visibility, scope, and lifetime)
  - Overview of type-checking
  - Garbage collection
- Abstraction mechanisms
  - Procedures, functions, and iterators as abstraction mechanisms
  - Parameterization mechanisms (reference vs. value)
  - Activation records and storage management
  - Type parameters and parameterized types
  - Modules in programming languages
- Object-oriented programming
  - Object-oriented design
  - Encapsulation and information-hiding
  - Separation of behavior and implementation
  - Inheritance
  - Polymorphism
  - Collection classes and iteration protocols
  - Internal representation of objects and method tables
- Functional programming
  - Overview and motivation of functional languages
  - Recursion over lists, natural numbers, trees, and other recursively-defined data
  - Pragmatics (debugging by divide and conquer; persistency of data structures)
  - Amortized efficiency for functional data structures
  - Closures and uses of functions as data (infinite sets, streams)

**Tentative schedule**

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Readings and assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 30</td>
<td>Introduction and course logistics</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Chapter and Section</td>
<td>Reading Notes</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>September 6</td>
<td>Chapter 1 Preliminaries (Sec 1.1-1.3)</td>
<td><strong>Reading:</strong> Sebesta 1.1-1.3</td>
</tr>
<tr>
<td></td>
<td>Chapter 2 Evolution of the Major Programming Languages</td>
<td><strong>Reading:</strong> Sebesta 1.4-1.8, Chap.2 and C Ref Chap. 1.</td>
</tr>
<tr>
<td>September 13</td>
<td>Chapter 3 Describing Syntax and Semantics</td>
<td><strong>Homework 1</strong> assigned. <strong>Reading:</strong> Sebesta Chap.3 and C Ref Chap. 8.</td>
</tr>
<tr>
<td></td>
<td>Chapter 4 Lexical and Syntax Analysis</td>
<td><strong>Homework 1</strong> due. (11:59pm, Sep. 20, <em>so on so forth</em>). <strong>Reading:</strong> Sebesta Chap. 4 and C Ref Chap. 2.</td>
</tr>
<tr>
<td>September 27</td>
<td>Chapter 5 Names, Bindings, and Scopes</td>
<td><strong>Homework 2</strong> assigned. <strong>Reading:</strong> Sebesta Chap. 5.</td>
</tr>
<tr>
<td>October 4</td>
<td><strong>Midterm</strong></td>
<td><strong>Exam Time:</strong> 06:30 PM-09:15 PM.</td>
</tr>
<tr>
<td>October 11</td>
<td>Chapter 6 Data Types</td>
<td><strong>Homework 3</strong> assigned. <strong>Reading:</strong> Sebesta Chap. 6 and C Ref Chap. 5 and 6.</td>
</tr>
<tr>
<td>October 18</td>
<td>Chapter 7 Expressions and Assignment Statements</td>
<td><strong>Homework 3</strong> due. <strong>Reading:</strong> Sebesta Chap. 7 and C Ref Chap. 7.</td>
</tr>
<tr>
<td>October 25</td>
<td>Chapter 8 Statement-Level Control Structures</td>
<td><strong>Homework 4</strong> assigned. <strong>Reading:</strong> Sebesta Chap. 8 and C Ref Chap. 8.</td>
</tr>
<tr>
<td>November 1</td>
<td>Chapter 9 Subprograms</td>
<td><strong>Homework 4</strong> due. <strong>Homework 5</strong> assigned. <strong>Reading:</strong> Sebesta Chap. 9 and C Ref Chap. 9.</td>
</tr>
<tr>
<td>November 8</td>
<td>Chapter 10 Implementing Subprograms Chapter 11 Abstract Data Types and Encapsulation Constructs</td>
<td><strong>Homework 5</strong> due. <strong>Reading:</strong> Sebesta Chap.s 10 and 11.</td>
</tr>
<tr>
<td>Date</td>
<td>Chapter Title</td>
<td>Notes</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>November 15</td>
<td>Chapter 12 Support for Object-Oriented Programming</td>
<td>Homework 6 assigned.</td>
</tr>
<tr>
<td>November 22</td>
<td>Chapter 13 Concurrency</td>
<td>Homework 6 due.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reading: Sebesta Chap. 12.</td>
</tr>
<tr>
<td>November 29</td>
<td>Chapter 14 Exception Handling and Event Handling</td>
<td>Homework 7 assigned.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reading: Sebesta Chap. 13.</td>
</tr>
<tr>
<td>December 6</td>
<td><strong>Final Exam</strong></td>
<td>Exam Time: 06:30 PM-09:15 PM.</td>
</tr>
</tbody>
</table>