<table>
<thead>
<tr>
<th>COURSE TITLE BLACKBOARD SITE</th>
<th>MCS 4643 – Comparative Programming Languages Summer 2012 – <a href="http://my.ltu.edu">http://my.ltu.edu</a> and select CRN 5456</th>
</tr>
</thead>
</table>
| INSTRUCTOR                   | Mohammed El-Bathy  
Adjunct Faculty at Math & Computer Science Department  
Office hours by appointment |
| SCHEDULE                     | May 16, 2012 – July 26, 2012  
Refer to [http://www.ltu.edu/registrars_office/calendar_final_exam.index.asp](http://www.ltu.edu/registrars_office/calendar_final_exam.index.asp) for the last date to withdraw and other important registration related information. |
| LEVEL/HOURS PREREQUISITE     | Graduate or Undergraduate Degree / 03 credit hours  
Admission / prerequisite requirements:  
Undergraduate level [MCS 2534](http://www.ltu.edu/registrars_office/calendar_final_exam.index.asp) Data Structures |
Authors web site [http://textbooks.elsevier.com/web/](http://textbooks.elsevier.com/web/)  
| ADDITIONAL RESOURCES         | LTU Online student resources: [http://www.ltu.edu/ltuonline/](http://www.ltu.edu/ltuonline/) |
| TECHNICAL SUPPORT            | Technical support for using Blackboard is provided by the Helpdesk, 248.204.2330 or [helpdesk@ltu.edu](mailto:helpdesk@ltu.edu). Send the Help Desk a form detailing any issues by clicking here [http://tinyurl.com/3yqrne](http://tinyurl.com/3yqrne). |
COURSE SCHEDULE FOR TRADITIONAL SEMESTER COURSES

This fully online course begins with a partial week online course orientation period to familiarize yourself with the online learning environment and to meet online or via the phone with your instructor. Each subsequent week starts on a Monday and ends on a Sunday.

<table>
<thead>
<tr>
<th>Dates</th>
<th>Modules</th>
<th>Topics / Readings</th>
<th>Instructional Activities</th>
</tr>
</thead>
</table>
| Prior to Semester Start and May 14 – May 20 | Module 0 | • Online Learning Orientation  
• Course Orientation  
• Overview of textbook                                                                 | • Course orientation  
• Instructor conversation  
• Individual pre-assessment                                                      |
| **Week of May 21 – May 27**   | Module 1 | **Introduction:**  
• Art of Language Design  
• Programming Language Spectrum  
• Compilation and Interpretation  
• Programming Environments                                                                 | • Read Chapter 1.0 - 1.5  
• Review Lecture Presentations  
• Practice Self Review #1  
• Bb Discussion #1 – Start  
• Essay Out  
• Group formation |
| **Week of May 28 – Jun 03**   | Module 2 | **Programming Languages Syntax:**  
• Overview of Compilation  
• Tokens & Regular Expressions  
• Context-Free Grammars  
• Derivations and Parse Trees                                                                 | • Read Chapter 1.6 – 2.1  
• Review Lecture Presentations  
• Practice Self Review #2  
• Bb Discussion #1 – End  
• HW 1 – Out |
| **Week of Jun 04 – Jun 10**   | Module 3 | **Names, Scopes & Bindings:**  
• The Notion of Binding Time  
• Object Lifetime  
• Storage Management  
• Scope Rules  
• Modules                                                                 | • Read Chapter 3.0, 3.3  
• Review Lecture Presentations  
• Practice Self Review #3  
• Bb Discussion #2 - Start  
• HW 1 – Due  
• Programming Problem 1 - Out |
| **Week of Jun 11 – Jun 17**   | Module 4 | **Names, Scopes & Bindings:**  
• Implementing Scope  
• Meaning of Names within a Scope  
• The Binding of Referencing Environments  
• Macro Expansion  
• Separate Compilation                                                                 | • Read Chapter 3.4 - 3.9  
• Review Lecture Presentations  
• Practice Self Review #4  
• Bb Discussion #2 - End  
• HW 2 – Out |
| **Week of Jun 18 – Jun 24**   | Module 5 | **Control Flow:**  
• Expression Evaluation  
• Expression initialization  
• Structured and Unstructured Flow  
• Sequencing                                                                 | • Read Chapter 6.0 – 6.3  
• Review Lecture Presentations  
• Practice Self Review #5  
• Bb Discussion #3 - Start  
• HW 2 – Due  
• Programming Problem 1 - Due |
| **Week of Jun 25 – Jul 01**   | Module 6 | **Control Flow:**  
• Selection  
• Iteration  
• Recursion  
• Non-determinacy                                                                 | • Read Chapter 6.4 – 6.8  
• Review Lecture Presentations  
• Practice Self Review #6  
• Bb Discussion #3 – End  
• HW 3 – Out |
<table>
<thead>
<tr>
<th>Dates</th>
<th>Modules</th>
<th>Topics / Readings</th>
<th>Instructional Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Programming Problem 2 Out</td>
</tr>
<tr>
<td>Week of</td>
<td>Module 7</td>
<td>Control Flow:</td>
<td>• Read Chapter 7.0 – 7.2</td>
</tr>
<tr>
<td>Jul 02 – Jul</td>
<td></td>
<td>• Type System</td>
<td>• Review Lecture Presentations</td>
</tr>
<tr>
<td>08</td>
<td></td>
<td>• Type Checking</td>
<td>• Practice Self Review #7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Bb Discussion #4 - Start</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• HW 3 - Due</td>
</tr>
<tr>
<td>Week of</td>
<td>Module 8</td>
<td>Data Types:</td>
<td>• Read Chapter 7.3 – 7.6</td>
</tr>
<tr>
<td>Jul 09 – Jul</td>
<td></td>
<td>• Records &amp; Variants</td>
<td>• Review Lecture Presentations</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>• Arrays</td>
<td>• Practice Self Review #8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Strings</td>
<td>• Bb Discussion #4 – End</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Sets</td>
<td>• HW 4 - Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Programming Problem 2 Due</td>
</tr>
<tr>
<td>Week of</td>
<td>Module 9</td>
<td>Data Types:</td>
<td>• Read Chapter 7.7 - 7.11</td>
</tr>
<tr>
<td>Jul 16 – Jul</td>
<td></td>
<td>• Pointers &amp; Recursive</td>
<td>• Review Lecture Presentations</td>
</tr>
<tr>
<td>22</td>
<td></td>
<td>• Dangling References</td>
<td>• Practice Self Review #9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Lists</td>
<td>• HW 4 - Due</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Equality Testing &amp; Assignment</td>
<td>• Essay - Due</td>
</tr>
<tr>
<td>Week of</td>
<td>Final Exams</td>
<td>· End of Course</td>
<td>• Course Evaluation</td>
</tr>
<tr>
<td>Jul 23 – Jul</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

STUDENT EVALUATION

The course has (4) four Blackboard Discussions, (4) four Homework, (1) One Essay, (2) two Programming Problems, and a Final. Letter grades are awarded based on the total number of points achieved. Points are deducted for late assignments.

<table>
<thead>
<tr>
<th>Assignments</th>
<th>Total Points</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Online Discussions – 60 pts each</td>
<td>240</td>
<td>24%</td>
</tr>
<tr>
<td>4 Homework – 60 pts each</td>
<td>240</td>
<td>24%</td>
</tr>
<tr>
<td>1 Essay</td>
<td>120</td>
<td>12%</td>
</tr>
<tr>
<td>2 Programming Problems – 60 pts each</td>
<td>200</td>
<td>20%</td>
</tr>
<tr>
<td>1 Final Exam</td>
<td>200</td>
<td>20%</td>
</tr>
<tr>
<td><strong>Total Points</strong></td>
<td><strong>1000</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class Points</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>96% and above</td>
<td>A</td>
</tr>
<tr>
<td>90% – 95.99%</td>
<td>A-</td>
</tr>
<tr>
<td>87% – 89.99%</td>
<td>B+</td>
</tr>
<tr>
<td>83% – 86.99%</td>
<td>B</td>
</tr>
<tr>
<td>80% – 82.99%</td>
<td>B-</td>
</tr>
<tr>
<td>77% – 79.99%</td>
<td>C+</td>
</tr>
</tbody>
</table>
EDUCATIONAL GOALS

Comparative Programming Languages course is an introduction to language design and implementation. It presents the theoretical underpinnings of programming languages. It provides an overview of core of the key paradigms used in developing modern programming languages; It introduces the implementation of different languages to provide an understanding of the relationship between a source program and its execution behavior.

Typical topics include: Programming Language paradigms, Programming Language Syntax, Programming Language Names, Scopes, and Bindings; Control Flow; Data Types; Subroutines and Control Abstraction; Data Abstraction and Object Orientation; Alternative Programming Models including: functional, Logical, Concurrency & Scripting Languages.

STUDENT LEARNING OBJECTIVES / OUTCOMES

Upon successful completion of the Comparative Programming Languages, student will be able to:

1. Discuss the fundamental principles in both the design and implementation of programming languages.
2. Present the differences between programming languages families.
3. Analyze the syntax or textual structure of programs
4. Describe the high levels structure of program with emphasis on names, the bindings of names to objects, and the scope rules that govern which binding are active at any given time.
5. Provide overview of the compilation process and describe how a compiler or interpreter determine the semantics, or the meaning of a program
6. Examine the storage management, subroutines, modules, and classes .
7. Examine data abstraction in the form of object orientation, characterized by an encapsulation mechanism, inheritance, and dynamic method dispatch
8. Explain the implementation of polymorphism, access control, closures, and multiple and mix inheritance in object oriented programming languages.
9. Discuss type systems and type checking, including the notions of equivalence, compatibility, and inference types.
10. Present a survey of high level-type constructors, including records and variant, arrays, Strings, sets, and pointers
11. Apply the design and implementation of programming languages in solving different industry problems.
12. Examine, compare and contrast the design and implementation of declarative and imperative programming languages.

PREREQUISITE SKILLS

- Student must have completed Basic Data Structure.
- Student must be able to write computer programs in C, C++, or Java
• Student must be willing and able to use MS Visual Studio C++ 2008 or 2010 for programming assignments.

INSTRUCTIONAL METHODS AND COURSE ORGANIZATION

Blackboard Learning Environment – Blackboard at my.ltu.edu contains the syllabus, all assignments, reading materials, streaming videos, narrated PowerPoint mini-lectures, podcasts, written lecture notes, chapter quizzes, links to Web resources, and discussion forums. You will submit all assignments via Blackboard, and are expected to participate regularly in discussion topics. Please take time to familiarize yourself with the organization of the Blackboard site. You will want to check the site frequently for announcements reminding you of new resources and upcoming assignments.

Student/Instructor Conversations – Students keep in touch with the instructor via e-mail messages, telephone conference calls, and IM conversations.

Self-Assessments – Pre- and post- self-assessment tools will help students measure their entering skills and progress during the course.

Required Reading – Textbook chapters should be read according to the schedule outlined in the syllabus. Chapters will be discussed online.

Publisher Web Site – A publisher web site at http://www.wiley.com/college/silberschatz includes instructional materials, PowerPoint slides, case studies, application exercises, and practice quizzes. You should make use of as many of these resources as you need to be successful.

Assignments – For each module, assignments will be posted on the Blackboard with due dates.

CLASS POLICIES AND EXPECTATIONS

I plan to offer you a valuable learning experience, and expect us to work together to achieve this goal. Here are some general expectations regarding this course:

• Each student has a LTU email account. If you wish to use a different email address for this course, please change your email address in Blackboard under “Blackboard Tools”, then “Personal Information” and send an email to me to store your email address in my directory.

• Readings, discussion forum participation, and written assignments must be completed according to the class schedule. It is important to contact the instructor as needed to discuss personal needs regarding course requirements and assignments.

• It is essential that all students actively contribute to the course objectives through their experiences and working knowledge.

• All assignments must be submitted on schedule, via Blackboard, and using Microsoft Office compatible software. If you need to submit an assignment via email, contact the instructor in advance.

• Assignments must be completed to an adequate standard to obtain a passing grade. Requirements for each assignment are detailed in this syllabus.

• Be prepared to log into Blackboard at least once each day. Please focus your online correspondence within the appropriate Blackboard discussion forums, so that your colleagues may learn from you.

• At midterm and at the end of the course, you will be invited to participate in a University evaluation of this course. Your feedback is important to the University, to LTU Online, and to me as an instructor, and I strongly encourage your participation in the evaluation process.
It is important for you as students to know what to expect from me as your instructor:

- I will be available to you via e-mail, Wimba classroom, and phone. I will promptly reply to your messages.
- I will maintain the Blackboard web site with current materials, and will resolve any content-related problems promptly as they are reported to me.
- I will send out a weekly e-mail update to all class members to guide upcoming work and remind you of assignment due dates.
- I will return all assignments to you promptly, and will include individualized comments and suggestions with each assignment.
- I will hold our personal written or verbal communications in confidence. I will not post any of your assignments for viewing by the class without requesting your approval in advance.
- I will treat all members of the class fairly, and will do my best to accommodate individual learning styles and special needs.
- If any of these points need clarification, or should special circumstances arise that require my assistance, please contact me so that we may discuss and resolve the matter.

PRACTICAL GUIDELINES FOR CLASS LOAD EXPECTATIONS

A three-credit course generally requires 10-12 hours per week of time commitment. Here are some practical guidelines to help schedule your time commitments for this online course:

- A 14-week semester (the Summer semester is compressed into 10 weeks) would require at least 140 hours of time commitment to successfully complete all readings, activities, assignments, and texts as described in this syllabus.
- You should reserve at least 6 hours per week to read the required textbook chapters and resources, participate in online discussions, review presentation materials, and work through online quizzes. This effort will total at least 84 hours over the course of the semester.
- You should organize your remaining time to roughly correspond with the point value of each major assignment. This means that you should plan to spend at least:
  - 8-9 hours preparing your case study review;
  - 24-40 hours working with your group on the three parts of your semester-long project;
  - 8-9 hours working on the various components of your reflective consolidation (final exam).

These guidelines may not reflect the actual amount of outside time that you – as a unique individual with your own learning style – will need to complete the course requirements. The number of hours each week will vary based on assignment due dates, so please plan ahead to insure that you schedule your academic, work, and personal time effectively. The following graphic may be used to guide you in planning your weekly course work to remain on schedule:
ASSIGNMENT DETAILS

Course assignments and evaluation criteria are detailed below. Please review these requirements carefully. See the section Academic Resources / Assessment Guidelines for information about assessment of written and oral presentations.

Details for all assignments are shown below. Please note that you should not submit any assignments to the Blackboard “Digital Drop Box.” All assignments are submitted using the Blackboard “Assignments” or “SafeAssign” function. Some assignments are also posted to the Blackboard Discussion Forum for student comments.

<table>
<thead>
<tr>
<th>Assignment Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overview</strong></td>
</tr>
<tr>
<td>This course features several Self-Review Exercises, Class Discussions, and Teamwork/Essay/Case Study, Homework, and Programming Problems/Projects activities.</td>
</tr>
</tbody>
</table>

**Self-Review Exercises:**

Each module contains self-review exercise that addresses important programming language concepts. These exercises are ungraded activities; they are designed to enable you to test your knowledge, get immediate feedback and gauge your understanding of material. These exercises also help prepare you for the homework, quizzes and exams. Some of the self-review exercises cannot be answered only from the material presented in their corresponding modules; these are additional teaching and learning opportunities.

**Class Discussions:**

- Class discussions are a fundamental part of individual student work (other individual work might include worksheets, essays, web assignments, etc.).
- Class discussions are intended to mimic discussions that take place in a brick and mortar classroom. Therefore, students are almost always expected to post a substantial initial response and to reply to contributions from other students and the instructor.
- **Your post and responses must be substantial and innovative contribution to the discussion**
  - The minimum expectation for each Class Discussion is to post one quality Main Post and at least two quality-responses to others (others include your classmates and/or me).
  - I will read each of your posts in these areas and will provide you with open-ended/ exploratory/ expanding responses.
  - It’s important that you to participate in each Class Discussion and Teamwork forum. Always proofread and spell check your posts.
  - There will be four (4) Bb discussions in the area of Programming Languages.
  - Each Class Discussion assignment is worth a total of 60 points (6% of the final grade). Earn up to 30 points for each initial post and up to 50 points for each response, depending on quality).

**Teamwork/Essay/Case Study:**

- **Your Essay must be substantial and innovative contribution**
  - Essays must be formatted according to APA Document formatting protocol and organized according to “How to Organize your Essay” document (listed under Course Information.)
  - There will be (1) teamwork/Essay/Case Study activity in the area of Programming Languages.
Teamwork/Essay/Case Study assignment is worth a total of 120 points (12% of the final grade.) Earn up to 60 points for your participation/effort and up to 60 points for the deliverable (composition of your document and PowerPoint developed by you and/or your team, depending on quality).

**Programming Assignments:**

- Emphasize the concepts presented in the course.
- There will be 2 Programming Assignments.
- Each programming assignment is worth 100 points (10% of the final grade)

**Due Dates**

- Teamwork/Essay/case study Activity appears in Week 1 and due in Week 9.
- Homework Assignments:

<table>
<thead>
<tr>
<th>Homework</th>
<th>Published Date</th>
<th>Due Date</th>
<th>Grade Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework 1</td>
<td>5/28/2012</td>
<td>6/10/2012</td>
<td>60 pts (6%)</td>
</tr>
<tr>
<td>Homework 2</td>
<td>6/11/2012</td>
<td>6/24/2012</td>
<td>60 pts (6%)</td>
</tr>
<tr>
<td>Homework 3</td>
<td>6/25/2012</td>
<td>7/08/2012</td>
<td>60 pts (6%)</td>
</tr>
<tr>
<td>Homework 4</td>
<td>7/09/2012</td>
<td>7/22/2012</td>
<td>60 pts (6%)</td>
</tr>
</tbody>
</table>

- Programming Assignments:

<table>
<thead>
<tr>
<th>Programming Assignment</th>
<th>Published Date</th>
<th>Due Date</th>
<th>Grade Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Program</td>
<td>6/04/2012</td>
<td>6/24/2012</td>
<td>60 pts (6%)</td>
</tr>
<tr>
<td>Second Program</td>
<td>6/25/2012</td>
<td>7/15/2012</td>
<td>60 pts (6%)</td>
</tr>
</tbody>
</table>

**Deliverables and Evaluation:**

- Essays must be formatted according to APA Document formatting protocol and organized according to "How to Organize your Essay" document (listed under Course Information.)
- Teamwork/Essays/case studies are submitted via Bb. Word, Excel or Journal format is preferred; however I will accept text (txt, rtf).
- Programming Problem assignments are submitted by zipping the Visual Studio project folder (after deleting the debug folder) and submitting via Bb assignments.

**SYLLABUS ADDENDA**

Please see the LTU Online “Current Students” web site [http://www.ltu.edu/ltuonline/] for comprehensive information about Lawrence Tech’s academic services, library services, student services, and academic integrity standards. The content of this web site is explicitly included in these syllabus requirements.

The LTU Online “Current Students” web site also includes grading rubrics used by your instructor to evaluate written assignments, discussion forum participation, and group assignments. Please note that the SafeAssign anti-plagiarism product will be used for written assignments submitted for this course. Please see the instructions included on the eHelp web site regarding the use of the SafeAssign product.
Undergraduates: Leadership Transcripts

The leadership transcript enables students to track co-curricular activities that are undertaken above and beyond the requirements of the LTU curriculum. The leadership transcript serves students by enhancing the leadership portfolio; providing the opportunity for a transcript of distinction; enhancing their resumes; and assisting in articulating leadership experience. It can be accessed by logging on to Banner Web and clicking the Student and Financial Aid tab. Leadership Activities is located at the bottom of the list.