COURSE: CMPS 344 - Programming Languages
Department of Computing Sciences, University of Scranton
(Class meets in LSC 091 on M-W-F from 10:00 AM - 10:50 AM)

DATE: Spring 2017 (January 30, 2017 - May 19, 2017)

INSTRUCTOR: P. M. Jackowitz
OFFICE: LSC 192
TELEPHONE: (570)941-6107
EMAIL: paul.jackowitz@scranton.edu
WWW: http://www.cs.scranton.edu/~jackowitz/

OFFICE HOURS: As posted, and by appointment.

Course Description: (Prerequisite: CMPS 240 [CMPS 250 Recommended]) A study of programming languages from both the theoretical and practical perspectives. A survey of major and developing paradigms and languages is undertaken which includes use of specific languages to broaden the student's experience. Implementation is studied through an introduction to language translation along with a study of run-time models and interfaces with underlying virtual machines. (Essentially from Undergraduate Catalog 2016-2017)

STUDENT LEARNING OUTCOMES: At the completion of this course the successful student will have:

1. Knowledge and understanding of the evolutionary history of programming languages and of computing in general.
2. Knowledge of and experience with formal languages, including regular and context-free grammars; specifically as they pertain to the lexical and syntactic aspects of programming languages.
3. Knowledge of and experience with Attribute Grammars as a means of representing semantic elements of programming languages.
4. An in depth understanding of the process of language translation.
5. An appreciation of the concept of "binding" and its use in understanding many aspects of programming and programming languages.
6. Facility with the terminology of programming languages regarding matters such as abstraction, data types, expressions, statements, control flow and subprograms.
7. Exposure to selected relevant constructs present in a variety of programming languages, presented as a means of appreciating the alternative approaches presented by different programming languages. Languages such as Ada, Eifel, Modula-2, etc. are typical.
8. Fluency in LISP programming as an example of the functional paradigm.
9. Exposure to the Prolog programming language as a means of appreciating the declarative or logic paradigm.

Course Web Site:
(This page serves as our primary electronic communication tool for this course. You will use it to access required and optional course material, submit assignments for grading, and receive feedback. You must register to gain access to the functionality of this site.)

GRADING:

<table>
<thead>
<tr>
<th>Tests: (approximate date)</th>
<th>Worth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week of March 6th</td>
<td>25%</td>
</tr>
<tr>
<td>Final Exam: (comprehensive, and yet to be scheduled)</td>
<td>25%</td>
</tr>
<tr>
<td>Quizzes: (as announced)</td>
<td>0%-10%</td>
</tr>
<tr>
<td>Assignments:</td>
<td></td>
</tr>
<tr>
<td>Programming, homework, etc.</td>
<td>40%</td>
</tr>
<tr>
<td>Project/Report/Presentation:</td>
<td>0%-10%</td>
</tr>
</tbody>
</table>

Attendance and Class Participation considered.
(Your attendance at all classes is expected. The accumulation of more than four absences may result in a diminished final course grade.)

PROCEDURES:

Lectures:
- please sit in the same seat for every class
- feel free to ask and answer questions, and to contribute to discussions

Tests and Quizzes:
- always announced in advance
- no make-ups will be given
- notice must be given if a test must be missed due to serious illness or emergency

Assignments:
- each student is required to do his/her own work
- discussions and cooperation among students is encouraged, but must not be to the point of representing someone else’s effort as your own
- academic dishonesty will be dealt with severely
- each assignment will have a specified due date, and a deadline (approximately 3 calendar days hence)
- work submitted after the due date is considered "late", will be accepted for grading but may be assessed a penalty (depending upon how late it is, and whether or not worthwhile preliminary work had been submitted by the due date).
- work may not be submitted after the deadline, is considered too late, may not be accepted for grading, and may receive a grade of zero (depending upon whether or not worthwhile preliminary work had been submitted by the deadline).
- incomplete work generally will receive a grade much higher than zero
- work not submitted will receive a grade of zero