CSC 430 – Programming Languages

1. CSC 430 – Programming Languages

2. credit units 4 contact hours 6

3. Course Coordinator: Aaron Keen / John Clements

4. Textbook (or other required material): None

5. a. Course Description:

Programming language design through evaluator implementation. Expressions, functions, environments, closures, mutation, objects, type systems, and syntactic abstraction. Syntactic, semantic, and static analysis properties. 3 lectures, 1 laboratory.

b. Prerequisite: CSC 349 and CSC/CPE 357.

c. Required/Elective/Selective Elective for CPE, CSC, EE, SE

<table>
<thead>
<tr>
<th></th>
<th>CSC</th>
<th>CPE</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Elective</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Selective Elective</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. a. Course Learning Objectives
The student will be able to:

• Implement software solutions of moderate complexity using a functional programming style.
• Identify syntactic elements of a programming language.
• Demonstrate parsing of textual input and transformation into abstract syntax trees.
• Demonstrate the use of a formal semantics of simple expressions, function calls, and mutation to compute the results of programs.
• Implement evaluators/interpreters to realize these same semantics.
• Explain the notion of scope and reason about the use of variables in arbitrary programs including the use of closures.
• Apply the rules of a type-checker to determine whether a program satisfies a type system.

b. Level at which Student Outcomes are addressed
(“B” = Basic level, “I” = Intermediate level, “A” = Advanced level)
7. Major Topics Covered: (number of lecture hours per)

- Functional programming style (5)
- Parsing and concrete syntax (2)
- Abstract Syntax. (2)
- Semantics. (9)
- Type Systems. (3)
- Scoping and Parameter Passing. (2)
- Closures. (2)
- Dynamic Memory Allocation and Garbage Collection. (2)