**Course Outcomes**

**CSC 2700**

**CSC 2700: Special Topics in Computer Science**

**“Programming in Python”**

**Credit Hours:** (1-3) variable credit

**Frequency:** on demand

**Prerequisites:**
CSC 1254 or CSC 1351 or permission of department.

**Prerequisites by Topics:**
Any 2-course sequence in an object-oriented programming language

**Catalog Course Description:**
May be taken for a max. of 6 hrs. when credit topics vary. Total credit earned in CSC 2700 and 4700 should not exceed 9 hours. Specialized areas of current interest in computer science.

**Class Title:** Programming in Python  (Spring 2007)

**Course Outcomes**
1. Master an understanding of scripting and the contributions of scripting languages.
2. Master an understanding of Python especially the object-oriented concepts,
3. Master an understanding of the built-in objects of Python,
4. Be exposed to advanced applications such as TCP/IP network programming, multithreaded programming, Web applications, discrete-event simulations, etc.

**Texts and Other Course Materials**

**Major Topics**
- Overview of scripting languages, the role of scripting languages and the appropriateness of these languages for particular applications; differences in terms of power, clarity, degree of object-orientation, etc.,
• Comparison of scripting languages with other object oriented languages,
• Portability issues,
• Syntax and semantics of the Python language including types, operators, function, modules, classes, exception handling,
• Built-in objects of numbers, strings, lists, tuples and dictionaries, iterators and generators,
• Regular expressions,
• Standard libraries for Python,
• Application interfacing,
• Interfaces to system languages,

Assignments/Projects/Laboratory Projects/Homework
A number of programming assignments required to demonstrate specific Python programming skills. Example program: Write a modular program in Python according to the specifications provided in the assignment.

Curriculum Category Content (estimated in semester hours)

<table>
<thead>
<tr>
<th>Area</th>
<th>Core</th>
<th>Advanced</th>
<th>Area</th>
<th>Core</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td>10</td>
<td>4</td>
<td>Data Structures</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Software Design</td>
<td>3</td>
<td>1</td>
<td>Prog. Languages</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Computer Arch.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Relationship to Criterion 3 Outcomes

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Math and Fundamentals:

Data Structures:
Sequences (lists, tuples, dictionaries) – 7 hours

Algorithms and Software:
Control structures – 3 hours
Tuples, objects – 5 hours,
Modules, functions, classes – 5 hours,
OS services – 4 hours
The Python library – 6 hours
Design – 4 hours use of classes, modules and functions in python
Computer Organization and Architecture:

Concepts of Programming Languages:
Coroutines (with respect to generators) – 1 hour,
Iterators – 2 hours,
Functional programming support (lambda, list comprehensions, generator expressions, map(), zip(), reduce(), filter() – 5 hours,
Threads – 2 hours

Social and Ethical Issues:

Oral Communication (presentations):

Written Communication:
All programming assignments must meet the criteria set forth in Guido’s “Style Guide for Python Code” found at:
http://www.python.org/dev/peps/pep-0008/
and “Docstring Conventions” at:
http://www.python.org/dev/peps/pep-0257/.

Course Coordinator: Leonard Blanks
Last Modified: May 8, 2007