

<b>Course #:</b>	<b>CSCI-C 335</b>
<b>Course Title:</b>	<b>Computer Structures</b>
<b>Course Type:</b>	Required core
<b>Prerequisites:</b>	P: C201 Computer Programming II.
<b>Credits:</b>	4
<b>Text Book:</b>	Textbook varies by instructor. Instructional materials distributed in class.
<b>References:</b>	Class notes
<b>Current Catalog Description:</b>	Computer architecture and machine language; internal data representation; symbolic coding and assembly systems; macros; program segmentation and linking; I/O devices; serial communication. Projects to illustrate basic machine structure and programming techniques.
<b>Course Goals</b>	<p>The student who completes this course:</p> <ol style="list-style-type: none"> <li>1. Will encounter and understand differing computer architectures.</li> <li>2. Will become familiar with multiple computer instruction sets.</li> <li>3. Will understand the basics data representation.</li> <li>4. Will gain assembly language proficiency by implementing a substantial project.</li> </ol>
<b>Major Topics Covered in the Course</b>	<ol style="list-style-type: none"> <li>1. Introductory digital logic elements, such as AND, OR, EOR, NOT, and their use in components such as adders and memory, as background to more abstract descriptions of computer structures and machine-level instruction representation.</li> <li>2. Abstract view of computer architecture: memory, CPU, BUS, I/O. The stored program. The fetch/execute cycle.</li> <li>3. Integer, float, and character representations. Relationship to data types in high-level languages. Loss of precision with floats. Finite ranges with integer.</li> <li>4. Logical, arithmetic, shift operations. Bit operations and masks. Relationship to operations provided in high-level languages.</li> <li>5. Memory as 1-dim array. Mapping high-level data structures to 1-dim memory array.</li> <li>6. Flow of control operations. Mapping high-level structured control to assembly-level branches.</li> <li>7. Registers. Addressing modes.</li> <li>8. Procedures, parameter passing, the runtime stack.</li> </ol>

	<p>[Calling assembly language subroutines from a high-level language.]</p> <ol style="list-style-type: none"> <li>9. The 2-pass assembly process. Assembler directives. Externals, linking, loading.</li> <li>10. Overview of I/O, communication between I/O and CPU.</li> <li>11. Interrupts and interrupt routines.</li> <li>12. Exception processing and error handling.</li> <li>13. Comparison of different architectures and instruction sets.</li> </ol>																		
<b>Laboratory projects (specify number of weeks on each)</b>																			
<b>Estimate Curriculum Category Content (Semester hours)</b>	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Area</th> <th>Core</th> <th>Advanced</th> </tr> </thead> <tbody> <tr> <td>Algorithms</td> <td>20</td> <td></td> </tr> <tr> <td>Software Design</td> <td>10</td> <td></td> </tr> <tr> <td>Comp. Arch.</td> <td>20</td> <td>5</td> </tr> <tr> <td>Data Structures</td> <td></td> <td></td> </tr> <tr> <td>Prog. Languages</td> <td>10</td> <td></td> </tr> </tbody> </table> <p>Additional hours may be dedicated to curriculum categories not listed above. For example explanation of concepts and theories. Discussion of social and ethical issues, discussion of interpersonal relationships and working within groups.</p>	Area	Core	Advanced	Algorithms	20		Software Design	10		Comp. Arch.	20	5	Data Structures			Prog. Languages	10	
Area	Core	Advanced																	
Algorithms	20																		
Software Design	10																		
Comp. Arch.	20	5																	
Data Structures																			
Prog. Languages	10																		
<b>Oral and Written Communications</b>	Not a course objective.																		
<b>Social and Ethical Issues</b>	Not a course objective.																		
<b>Theoretical Content</b>	<ul style="list-style-type: none"> <li>• Arithmetic</li> <li>• Data representation</li> <li>• Machine design</li> <li>• Comparative instruction sets</li> <li>• Interpreter development 4hr.</li> </ul>																		
<b>Problem Analysis</b>	Interpreter development, CPU architecture considerations. 4hr.																		
<b>Solution Design</b>	Not a course objective.																		
<b>Prepared By</b>	Wolfer																		