I. Course Title: Computer Science I: Fundamentals of Programming

II. Course Prefix/Number: CSCI 207

III. Credit Hours: 3

IV. Prerequisite(s): ACT mathematics score of 19 or more or Math 099 with a grade of S or Math 104 with a minimum grade of C- or appropriate score on Mathematics Placement Test.

V. Catalog Description: This course is the first course in the introductory programming sequence for computer science and business administration/management information science. Through the study of object design this course introduces the basics of graphical user interfaces, data types, control structures, methods, arrays, strings, and files.

VI. Curricular Relationships: This course is the first core course in the B.S. Mathematics degree with an emphasis in Computer Science. It is also required for the B.S Business Administration/Management Information Systems (MIS) degree. It is a requirement for a minor in Communications Technology and a minor in Computer Science. It can be applied toward an Associate of Science Degree in Computer Security. It is delivered at a level suitable for non-majors.

VII. Student Learning Outcomes:

- Students will know how to develop well documented programs for solving problems using sound object design and structured programming;
- Students will understand the concepts of class, object, instance variables, methods and parameter passing;
- Students will understand the concepts of variables, data types, expressions and assignment;
- Students will know how to import class libraries/packages, particularly packages for graphics and GUI applications;
- Students will know how to program with fundamental control structures, selection and iteration;
- Students will know the fundamental use of strings and arrays, as well as fundamental algorithms used with these data structures;
- Students will demonstrate problem solving proficiency at the appropriate level as it applies to basic software development;
- Students will know how to use files and streams for input/output in simple programs.
- Students will be able to write programs that implement proper exception handling techniques.
VIII. Content Outline:
- Basic object-oriented concepts for implementing proper program design.
- Fundamentals: implementing objects as classes, reference variables, language environment, basic syntax, object properties, methods, and parameter passing.
- Primitive data types, expressions and returning values from method calls.
- Variable scope
- Class libraries and packages.
- The fundamental structure of a GUI program and simple GUI components.
- Flow of control constructs: selection and iteration.
- Fundamental data structures: strings and arrays.
- Basic 2D graphics.
- Debugging techniques and tools.
- Fundamentals of event driven programming
- The use of streams input/output processing.
- Basic testing and debugging techniques including an introduction to unit testing.
- Exception handling.
- Documentation: specification (what module/program does), implementation (how it is done) and variable/parameter usage.

IX. Course Procedures/Policies/Grading Scale:
- Lecture/presentation/discussion.
- Students will design and implement 10 – 18 small programs.
- Students will work exercises to become proficient with underlying concepts and techniques.
- Students will take 2-3 exams during the semester and a comprehensive final.
- Exams are approximately 50% of the term grade and programs approximately 50%

X. Required/Recommended Readings:

Example textbooks include:

XI. Issues Unique to Course: This course requires additional computer laboratory time.

The course will meet five days per week for the first 12 weeks of the semester. A one-hour comprehensive final exam will be given on the last day of the course. Students interested in continuing their study of programming will be expected to continue on to CSCI 208 or BUS 331.
XII. Additional Departmental Issues:

This course addresses the topics listed in the CS body of knowledge core as described in the ACM/IEEE document *Computer Science Curriculum 2008: An Interim Revision of CS 2001, Report from the Interim Review Task Force*. The course includes topics from the following core areas: PF (Programming Fundamentals), SE (Software Engineering), AL (Algorithms and Complexity), HC (Human-Computer Interaction), GV (Graphics and Visual Computing), and PL (Programming Languages).

<table>
<thead>
<tr>
<th>Core Area</th>
<th>Topic</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF</td>
<td>Fundamental Constructs</td>
<td>9/9</td>
</tr>
<tr>
<td>PF</td>
<td>Algorithmic Problem Solving</td>
<td>3/6</td>
</tr>
<tr>
<td>PF</td>
<td>Data Structures</td>
<td>3/10</td>
</tr>
<tr>
<td>PF</td>
<td>Object Oriented</td>
<td>2/8</td>
</tr>
<tr>
<td>PF</td>
<td>Event Driven Programming</td>
<td>2/4</td>
</tr>
<tr>
<td>AL</td>
<td>Basic Analysis</td>
<td>1/4</td>
</tr>
<tr>
<td>PL</td>
<td>Virtual Machines</td>
<td>1/1</td>
</tr>
<tr>
<td>PL</td>
<td>Basic Language Translation</td>
<td>1/2</td>
</tr>
<tr>
<td>PL</td>
<td>Declarations and Types</td>
<td>1/3</td>
</tr>
<tr>
<td>PL</td>
<td>Object Oriented Programming</td>
<td>1/10</td>
</tr>
<tr>
<td>HC</td>
<td>Foundations</td>
<td>6/6</td>
</tr>
<tr>
<td>HC</td>
<td>Building GUI Interfaces</td>
<td>1/2</td>
</tr>
<tr>
<td>GV</td>
<td>Fundamental Techniques</td>
<td>1/2</td>
</tr>
<tr>
<td>SE</td>
<td>Software design</td>
<td>1/8</td>
</tr>
<tr>
<td>SE</td>
<td>Using APIs</td>
<td>1/5</td>
</tr>
<tr>
<td>SE</td>
<td>Tools and Environments</td>
<td>2/3</td>
</tr>
<tr>
<td>Electives</td>
<td></td>
<td>9</td>
</tr>
</tbody>
</table>