I. Course Title: Computer Science I: Object-Oriented Programming

II. Course Prefix/Number: CSCI 208

III. Credit Hours: 1

IV. Prerequisite(s): Completion of CSCI 207 with a minimum grade of C-.

V. Catalog Description: This course is the second course in the introductory programming sequence for computer science. The course emphasizes object-oriented programming through a study of encapsulation, polymorphism, coupling, cohesion, composition, aggregation, object-oriented design, interfaces, access control, and inheritance. Course content also includes an introduction to lists, wrappers, abstract, and final classes as well as the classic model/view/controller and singleton design patterns.

VI. Curricular Relationships:
This course is a continuation of Computer Science I. It is required for a B.S. Mathematics degree with an emphasis in Computer Science. It is a requirement for a minor in Communications Technology and a minor in Computer Science. It is a recommended course for anyone interested in developing a strong understanding of object-oriented design.

VII. Student Learning Outcomes:

- Students will effectively use dynamic lists for storage;
- Students will understand wrapper classes and the fundamentals of auto boxing/unboxing.
- Students will understand and effectively implement abstract classes.
- Students will understand and effectively implement final classes.
- Student will understand the concepts of encapsulation, inheritance, interfaces, composition, cohesion, coupling, aggregation, polymorphism, coupling, cohesion, and access control and be able to use the concepts to write well-designed object-oriented code;
- Students will understand and effectively implement model/view/controller architecture and the singleton design pattern.
VIII. Content Outline:
- Object-oriented concepts: inheritance, composition, aggregation, polymorphism, interfaces, encapsulation, cohesion, coupling, and access control.
- Abstract classes and methods.
- Final classes and methods
- Dynamic Lists
- Wrapper classes, auto-boxing, and unboxing.
- Introduction to design patterns: model/viewer/controller architecture and singleton.

IX. Course Procedures/Policies/Grading Scale:
- Lecture/presentation/discussion.
- Students will design and implement 4 – 8 programs.
- Students will work exercises to become proficient with underlying concepts and techniques.
- Students will take one exam 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 last four weeks of the semester in which it is scheduled. Students are expected to take CSCI 207 during the first twelve weeks of the semester.

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), and PL (Programming Languages).

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Topic</th>
<th>Core Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF</td>
<td>Data Structures</td>
<td>2 (out of 10)</td>
</tr>
<tr>
<td>PF</td>
<td>Object Oriented</td>
<td>4 (out of 8)</td>
</tr>
<tr>
<td>PL</td>
<td>Abstraction Mechanisms</td>
<td>3 (out of 3)</td>
</tr>
<tr>
<td>PL</td>
<td>Object oriented programming</td>
<td>4 (out of 10)</td>
</tr>
<tr>
<td>SE</td>
<td>Software design</td>
<td>2 (out of 8)</td>
</tr>
</tbody>
</table>