#Course Outcome Guide (COG)

**Course:** CSCI 127 – Beginning Java/J++  
**Credits:** 3  
**Instructor:** TBD  
**Course Description:** Introduction to programming in the Java/J++ language

##Concepts and Issues
- Design Language
- Object-oriented design principles
- Basic control structures
- User interface, controller, and application logic layers
- Test plans

##Intended Outcomes

<table>
<thead>
<tr>
<th>Course</th>
<th>General Education or Program</th>
<th>Institutional</th>
</tr>
</thead>
</table>
| Create UML class diagrams  
Create classes from class diagrams  
Use simple design patterns  
Create unit tests using JUnit  
Use inheritance from interfaces and abstract classes  
Use proper Java exception handling techniques  
Use the various collection classes  
Use Java’s IO system  
Write a simple client and server program using networking classes  
Create Java packages  
Use threadsS | 1. Mathematics—including numeration literacy and the knowledge and use of statistical and logical processes.  
2. Analytical-gathering, organizing, and evaluating information  
3. Analogical-using former knowledge to help comprehend and explain new situations  
4. Critical Thinking—the ability to identify ad define criteria, understand biases, and construct objective judgments.  
5. Problem solving—the ability to analyze situations and synthesize solutions. | 1. Students will demonstrate effective communication skills.  
2. Students will use reasoning skills to analyze and solve problems. |

##Process Skills
- Apply object-oriented design to small software projects.
- Produce simple object-oriented programs demonstrating use of class definition, methods, primitive and reference data types, alternation and repetition control structures, and file-based and interactive input/output.
- Produce simple event-driven object-oriented programs using basic Java library components.
- Assess the quality of programs using simple glass box and black box testing strategies.
- Describe and demonstrate different physical data representations for primitive data types.
- Use good software development principles including object-oriented design, test planning and adherence to style guidelines.
Describe the purpose and operation of Java software development tools including compilers, editors, and integrated development environments; use tools to do software development.
Describe the Java runtime environments.
Identify and describe the activities involved in the software development process.