Computer Science Department
Masters Program Assessment

The faculty and staff of the Computer Science Department are dedicated to training and preparing students to become computer scientists. We believe that computer science has been and will continue to contribute greatly to all disciplines of study and world economy. We provide a high quality education to prepare students for productive careers in industry, academia, and government in a nourishing environment for teaching, learning, and research in the theory and applications of computing. This training is offered under the direction of the Graduate School at Western Washington University.

The graduate program in the Computer Science Department involves students in courses on computer science theory, concepts, and practice and in research projects. Students graduating from our program will have the ability to apply computing knowledge and mathematics to real world problems. They will be able to analyze problems and identify the computing requirements appropriate to its solution. Students will master the design-implementation-evaluation cycle for computer-based systems, processes, components, or programs to meet desired goals. Our program provides students with an understanding of professional, ethical, legal, security and social issues and responsibilities.

Goals:
1. To graduate students with strong, in-depth background in computer science
2. To prepare students for Ph.D. programs and advanced careers in information technology
3. To engage students in high quality research-oriented projects

Objectives:
1. To educate students in core concepts in computer science
2. To provide students with in depth knowledge, skills, and experiences in computer science
3. To focus on the applied aspects of computer science, especially real-world projects that use core concepts and expert knowledge of computer science
4. To provides students and faculty opportunities to engage in research in computer science

Assessment Methods
The Computer Science Department uses course-embedded assessment. Each of the classes taught has a list of course outcomes. Each class is assessed by the faculty member teaching that class to measure the course outcomes for that class. Each class will be assessed once for every three times the class has been taught.

Feedback and Program Improvement
At the end of each quarter, individual class assessment reports are submitted to the Graduate Committee of the department. In Spring quarter, the Graduate Committee will prepare a report using all class assessment reports. This report will be submitted to the department.

After the assessment report is completed, the Graduate committee will discuss any issues resulting from the assessment and prepare a report detailing any proposed actions to improve the curriculum. This report is submitted to the department for approval.
Course Outcomes

These are course outcomes for a majority of the CSCI graduate classes taught in the 2015-2016 school year. These course outcomes are used for the classes assessed during Winter and Spring 2016.

CSCI 509 – Operating System Internals
On completion of this course, students will demonstrate:
1. A thorough understanding of process scheduling and the system implementation of multi-thread applications.
2. A thorough understanding of memory management techniques in operating systems.
3. A thorough understanding of the implementation of file systems and device control.
4. A basic understanding of distributed system management.

CSCI 510 – Automata and Formal Language Theory
On completion of this course, students will demonstrate:
1. A thorough understanding of fundamental classes of languages including: regular, context free and recursively enumerable.
2. A thorough understanding of fundamental classes of machines including: deterministic and non-deterministic finite automata, pushdown automata, and Turing machines.
3. A thorough understanding of proof techniques (e.g. pumping lemmas, reductions) to classify languages.
4. A thorough understanding of Undecidability.
5. A thorough understanding of NP-Completeness.

CSCI 511 – Analysis of Algorithms
On completion of this course, students will demonstrate:
1. A thorough understanding of mathematical techniques used to determine time and space complexity of algorithms.
2. A thorough understanding of a variety of graph algorithms and matrix algorithms.
3. A thorough understanding of dynamic programming and greedy algorithms.
4. An ability to formulate efficient, effective solutions for various problems and justify the correctness and complexity of those solutions.

CSCI 512 – Design and Implementation of Computer Programming Languages
On completion of this course, students will demonstrate:
1. A thorough understanding of the syntactic specification of programming languages.
2. A thorough understanding of operational semantics of programming languages.
3. A thorough understanding of denotational semantics of programming languages.
4. An understanding of the pragmatics of programming language implementation.

CSCI 515 – Parallel Computation
On completion of this course, students will demonstrate:
1. A thorough understanding of parallel machine architecture.
2. A thorough understanding of parallel programming techniques.
3. A basic understanding of parallel languages.
4. The ability to design and implement programs in a parallel environment.
CSCI 571 – Machine Learning Algorithms
On completion of the course, students will demonstrate:
1. A thorough understanding of classification, regression and clustering.
3. A basic understanding of assorted, advanced machine learning algorithms.
4. A thorough understanding of one advanced machine learning algorithm.
5. A basic understanding of feature selection and transformation.
6. The ability to implement standard machine learning algorithms and run machine learning experiments.

CSCI 573 – Computational Linguistics
On completion of this course, students will demonstrate:
1. A basic understanding of natural language processing (NLP) concepts and tasks such as segmentation, spelling correction, part-of-speech tagging, parsing, semantic role labeling, text categorization, sentiment analysis of natural languages.
2. A basic understanding of corpus-based algorithms in natural language processing.
3. The ability to apply corpus-based language processing and simple machine learning algorithms.
4. A basic understanding of the limitations and promise of NLP.
5. The ability to understand research papers on NLP.
6. A thorough understanding of one selected, advanced topic in NLP.

CSCI 578 – Cryptography
On completion of this course, students will demonstrate:
1. A thorough understanding of the principles and inner workings of modern cryptography.
2. A thorough understanding of the weakness in cryptographic algorithms and protocols.
3. The ability to implement sound cryptosystems.

CSCI 579 – Spoken Language Processing
On completion of this course, students will demonstrate:
1. A basic understanding of human speech perception and production.
2. A basic understanding of assorted, advanced topics in spoken language processing.
3. The ability to build basic spoken language processing systems.
4. A thorough understanding of fundamental models and algorithms in spoken language processing.
5. A thorough understanding of how spoken language processing systems work.
6. A thorough understanding of one selected, advanced topics in spoken language processing.

CSCI 580 – Advanced Computer Graphics
On completion of this course, students will demonstrate:
1. A basic understanding of hardware system architecture for computer graphics.
3. A thorough understanding of shading models.
4. A thorough understanding of surface modeling.
5. A basic understanding of physically based rendering algorithms (raytracing, global illumination).

CSCI 601, 602, 603 – Research Experience
On completion of this course sequence, students will:
1. Have a deep understanding of one area of Computer Science.
2. Have written a paper about the research conducted in this sequence.
3. Have submitted the paper to a conference or a journal or will have identified a future conference to which the student will submit the paper when submissions are possible.

**Course Outcomes to Program Objectives Mapping**

<table>
<thead>
<tr>
<th>Course Outcomes to Program Objectives</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 509</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>CSCI 510</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSCI 511</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>CSCI 512</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSCI 515</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>CSCI 571</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSCI 573</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>CSCI 578</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSCI 579</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>CSCI 580</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSCI 601, 602, 603</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>