Program Review

Computer Science

Contributors

Full-Time Faculty

Goble, Colin
SY TCB 312

Iyer, Gayathri
RC 2 230

Jones, Doug
SY TCB 312

Liang, Li
SY TCB 312

Morales, Walter
RC 2 230

Part-Time Faculty

Abramson, Jeff

Carpenter, Ralph

Trigoboff, Michael

Table of Contents

Executive Summary

Enrollment in Computer Science programs is increasing at the Community College and University levels, and the employment outlook for graduates with 4-year or 6-year degrees is robust. PCC chartered the Computer Science program to prepare students for transfer to four-year institutions in Oregon, and our Program has enjoyed commensurate increases in enrollments and challenges while fulfilling our charter. This document discusses the performance of the Program, our effectiveness in preparing students for upper division studies, issues we have overcome, and opportunities we still must exploit in order to ensure that our students achieve their academic goals and are superbly prepared to further their education after leaving PCC.

Our formal transfer partners are Portland State University, Oregon Institute of Technology, Oregon State University, and the University of Oregon. While written agreements are in place only with these institutions, our students may transfer to virtually any upper-division Computer Science program with relatively minor alterations in their course selections while at PCC.

Nationwide enrollment in Computer Science programs is increasing and most classes at PCC are full and have waiting lists. The Computing Research Association notes "New enrollment in North American computer science and engineering programs rose 8 percent during the 2007-08 school years" and PCC's Computer Science enrollments have significantly increased the last few terms:

- CS program posted a 51.2% increase in FTE enrollments of CS students since 2005, and a 15% increase from 07-08 to 08-09.
- Most of our enrollments are in the introductory classes of CS160, 161, and 162 (Programming in C++), and the Gaming class, CS 133G.
- Enrollments in Distance Learning classes and on-campus sections are equally balanced in the year 08-09, with a recent uptrend in on-campus learning.

We have been fortunate to have full time faculty presence in both campuses and maintain our pool of very qualified adjunct instructors. Some of them have been teaching at PCC for over 10 years. Many of them do work in the computer industry, such as Intel, Tektronix and teach our high level courses allowing a direct connection with our transfer students offering them valuable advising about current trends in the industry.

Rock Creek added a full time instructor in the last 3 years and that allowed a better balance between PT/FT ratios. This winter term (2010) 52% of our courses are taught by FT instructors to the benefit of the students. We have improved tremendously in our teaching methods by using the hands on philosophy. Most of our classes are lab oriented and are taught in computer equipped classrooms. We have updated much of our introductory curriculum to be more engaging for entry-level students with a hands-on focus including interactive games and robotics.

The CS department continues to work with PSU, OSU, UO, and OIT to renew transfer agreements, so students can transfer seamlessly to these institutions and finish their degrees. For instance, PSU recently revised their CS transfer requirements, resulting in the revision of many CS courses.

CS has inactivated 2 courses, and added 4 new courses and updated the content of the remainder of its courses in the last 3 years. The CS program at PCC offers all the classes online except CS 250 and CS 251, both of which are new in this academic year.

The requirement for training conceptual age cognitive and creative skills appears to be long term industry need. So, PCC's CS program has introduced 2 new gaming courses CS 133G and CS 233G with

outstanding success. Our instructors also incorporate gaming assignments in regular classes, thereby emphasizing cognitive skills along with implementation skills.

Our program must evolve to meet the needs of students and community. To strengthen the program, we intend to make improvements in the following areas: Instructional technology, Course Delivery, Faculty, Curriculum, and Process.

Introduction and Goals

State of the Discipline

The Computer Science program emphasizes transferability to other institutions. According to a recent survey, winter term 2010, PSU continues to be the main choice of our transfer students. However, we also see the need to support students from allied programs such as CIS (Computer Information Systems) and other engineering disciplines with our 100 level classes. In addition, the CS department offers courses for professional development and academic advancement to those in the industry.

Student Survey, Winter Term 2010

189 students responded to the survey question, "To which school do you plan to transfer?"

Portland State University	**80**	(42%)
OIT	**16**	(8%)
Oregon State University	**8**	(4%)
University of Oregon	**3**	(2%)
Other	**15**	(8%)
No Answer	**67**	(35%)

Employment Projections

The CS department monitors employment projections, which are very encouraging. According to the Bureau of Labor Statistics (http://www.bls.gov/oco/ocos304.htm)

> "Computer scientists held about 28,900 jobs in 2008. Although they are increasingly employed in every sector of the economy, the greatest concentration of these workers, about 23 percent, is in the computer systems design and related services industry. Many computer scientists are also employed by software publishing firms, scientific research and development organizations, and in education."

Employment projections for Computer Science graduates show increasing demand for these skills, and PCC is in a position to support the growing demand for Computer Science education. Detailed information on employment trends is provided as an Appendix to this document.

We closely watch the industry trends, student's feedback, and offer classes (when resources are available) to supply that new demand. New offerings such as two gaming classes and introduction of graphical programming tools such as Ceebot, darkGDK, and Gamemaker are very popular in attracting students to the computer science program.

Program Goals

The program goals are presented as a vision for our program in several different areas and its integration with the college's Mission, Values and Goals:

1. Community Spirit: An important goal is to make the district-wide Computer Science department as much of a community as possible.
    - The CS Department has observed that student motivations are higher when they are around a group of people outside of class working on the same projects as they are, and taking similar classes.

    There are several concrete steps to be taken to accomplish this goal:

    - Provide a place where students can study and can work in groups, since the department emphasizes group learning and group projects.
    - Provide students with on campus and online tutors to foster student's success and integration with PCC's services and community.

2. Program: It would be ideal to work towards building up the CS program infrastructure. The department is charged by the community college with providing computer science classes that are equivalent to their college and university counterparts. Since our computer science classes are transfer classes, and since our program is a transfer program, it is vital to our continued credibility at these institutions that our program is of a good quality. Our faculty must take extra time to be familiar with changes to the technology because computer science is continually evolving at a high rate. At the same time, our discipline is more than a technology. As the name implies, ours is a science, and because we are a science, it is important that we teach the complex concepts that are the core of our discipline. These concepts remain steady even as the technology in the discipline changes. Thus, students who learn computer science from us today will be able to understand changes in languages, operating system, and other elements of the technology that will inevitably occur as time goes by. In addition, since ours is such a relatively young discipline, there are many changes occurring in the pedagogy of computer science. Professional development support to allow faculty train on these new technologies is essential for the student's success and ours. Our students, for most part, are in many cases a step ahead when dealing with the use of Operating Systems, as an example. Many students already have Windows 7 while we are still working with XP. Many different approaches to the material are being explored and it is natural that some of these approaches are showing much more promise than others such as the introduction of graphical software development tools in some of our classes. It is important that we stay current in this evolving debate, so that we can use the best approaches to teach an admittedly difficult subject.

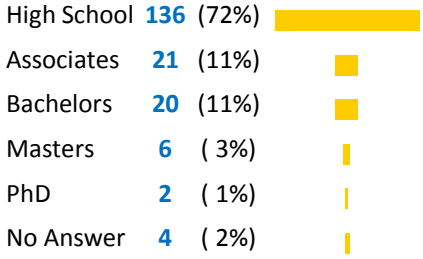
3. Pedagogy: Keeping current in the technology of our field and in computer science pedagogy takes a great deal of time and effort. It is therefore important that we have the resources necessary to carry out our charter. These resources include proper equipment and auxiliary help. A good infrastructure contains several components:
    - Many of the current approaches to computer science education adopt a "hands on" philosophy, which requires a laboratory to carry out related labs. Some of our courses are taught in classrooms with no computers. Because the students can show their programs to anyone in a class/lab environment, it becomes a wonderful way to motivate students when they are learning the difficult skill of complex programming and share their results with other students.

- Online offerings are very popular, but unfortunately, there are no tutors to support those students. It is essential for some of those students to get that extra support, so it is one of our goals to advocate for a tutor for the online students.

4. Program Positioning and Marketing: We would like to sell our program as a wonderful, thorough, personal, and inexpensive first 2 years of a four-year program and as preparation for graduate school for those who already hold a bachelor's degree in another discipline. We believe that the community undervalues the tremendous strength of our CS program. Our program has instructors of equal or better quality than that of many institutions for the first two years of a computer science program at a 4-year institution. Additionally, our faculty has a strong desire to teach. This combination makes us a competitive choice for students going into a computer science 4-year institution. Most universities have more than 50 students per class in their beginning level computer science classes. These students are all hoping to be majors in the computer science program. Community colleges should be seeing a similar growth in their own computer science programs, as long as the students see their programs as equivalent to our program. As it continues to grow in quality, we will attract many of the students from universities who feel more comfortable in an intimate setting with economic advantages.

   Our survey (winter, 2010) respondents indicated that many hold HS diplomas. Approaching HS counselors before the students' transition to college will improve the advising process and their success in college, in accordance with PCC's Student's Success goals. One of our goals is to have a good marketing package to provide to counselors, outreach coordinators, High school and PCC advisors to inform students about PCC's CS program.

(Survey Winter 10) Describe your educational background



5. Service Component: We are successful with "service" courses that meet the needs of industry professionals and students from other disciplines. The implementation of more service courses addressed toward this market would increase the number of students that we currently have and it could invite newer students to our program. A limited number of rooms equipped with computers keep us from offering more of these courses, so we would like to equip more classrooms with computers. We also feel it is important for us to have computer equipment that is of the same quality (state-of-the-art) as the businesses in the community. We would like to seek internal opportunities and partnerships with other PCC's disciplines that may use some of our classes in our program avoiding duplicate use of resources. Our student population come to us with various backgrounds and needs as the respondents showed in this question. We will focus on our students diverse needs focusing

our efforts on our transfer students and seek other opportunities, cultivating partnerships and access to the community.

(Survey Winter 10) Describe the reason that most accurately describes your reasons for taking this class

I am working on a BS in CS degree and plan to transfer.	**65**	(34%)
I have a degree and I need this class for graduate work.	**8**	(4%)
Working towards a degree and this class is required.	**38**	(20%)
I am taking this class for professional development.	**15**	(8%)
I am thinking about a career in the computer industry.	**40**	(21%)
Other	**15**	(8%)
No Answer	**8**	(4%)

6. Transferability: The transferability of courses to four-year Computer Science programs at universities and colleges is the charter of the Computer Science program. As we revise the dual enrollment programs and increase the number of transfer agreements with other schools in Oregon (and perhaps abroad), we will be able to increase the number of current class offerings and number of students in those classes. This charter makes it imperative that our classes be comparable to analogous classes at four-year institutions. Other institutions should feel confident in allowing students to transfer courses they have taken at PCC to their institution. It is our goal to work closely with the computer science faculty on the other PCC campuses to be sure that our classes continue to be of high quality. As we have discussed at computer science SACC meetings, the synergy among our faculty is the most important component of our department. Computer science is such a quickly evolving field that no one of us can be current in all aspects of it in its entirety. Instead, it is important that all of us work to stay current in part of the field and important that we share our expertise with each other. All the members of the computer science discipline at PCC feel that it enhances our teaching and overall understanding when we work with the other faculty in our discipline. We learn from each other about ways of teaching methods to best help our students, and what is occurring in our rapidly changing field.

7. Lab Facilities: We would like to have access to more computer-equipped classrooms and offer more "hands on: classes, as this seems to be a learning style that our students and instructors both prefer and approve.

8. Industry Partnerships: We want to create partnerships with industries in our region, and use their facilities or ours to offer service courses that answer the needs of the community. This will not replace our core computer science curriculum, since this core curriculum is our charter as a department. Instead, it will augment our core curriculum where possible.

9. Academic Partnerships: We would like to have a closer contact with computer science departments at the universities and colleges where our students transfer. We already

regularly attend meetings with these academic bodies, but we would like to become more integrated with these institutions. One way to accomplish this goal would be collaborating on research projects with the faculty at these institutions. This would further our relationships with the schools and allow them to see the quality of our best students. It would also give our students a chance to begin doing research quite early in their academic career, giving them a 'leg up' when they apply to these school s' computer science program. As an added benefit, it would allow our faculty to stay current in their field. Another venue of marketing and partnership is to offer classes at local high schools. This experiment has been done at some local schools and will be done again throughout the district.

10. Distance Learning: We would like to increase the number of distance learning courses that is offer, so we can reach those students who are willing to take our courses but who do not have the time to come to our campus. In addition, as a way of leveraging some current faculty visits to foreign universities that offer computer related courses. This effort can collaborate with PCC's internationalization campaign. We would like to start relationships with foreign universities. After the completion of the main core courses, students will come to the US to finish their four-year education in one of the state schools. This will reduce the student's cost of education, since many foreign students cannot attend American universities due to the high cost of living required to study abroad for a full four years. Some responses in our survey show the needs of our online students and the need to balance on campus/online offerings.

(Survey Winter 10) Have you taken any distance learning courses (check all that apply)

Through PCC	98	(52%)
At another school	34	(18%)
Never	74	(39%)

Would you take a CS class online

Yes	97	(51%)
No	40	(21%)
Maybe	51	(27%)
No Answer	1	(1%)

Would you consider finishing the whole CS program online

Yes	54	(29%)
No	87	(46%)
Maybe	47	(25%)
No Answer	1	(1%)

11. Night and Saturday Offerings: We would like to offer more classes on Saturdays. We have coordinated with, and will continue scheduling classes collaboratively around, the

district providing access. This allows students to take our classes at either campus and at convenient times. The maps included in the Appendices show our current student population and perhaps the need to offer some of our popular courses near their homes. The maps use census tracts, the smallest territorial unit for which data are available. The darker the colors, the larger the concentration of CS students.

12. Women and Minorities: We need to continue to attract more women and minority students to Computer Science collaborating with the CAMP program or with the student leaders. We would like to become more of an entry point into computer science for non-traditional students. These students can benefit greatly from what we at PCC have to offer. We offer one-on-one instruction with knowledgeable faculty who can give our students the little bit of extra time, patience, and encouragement they need to be successful in a computer science program. We also want to continue to seek scholarships for these and other students from government or private sources, since many are unaware of the scholarships.

(Winter 09) Student Survey

Gender			
Male	**156**	(83%)	
Female	**31**	(16%)	
No Answer	**2**	(1%)	

13. Gaming: This topic has been a pressing issue for two years now. The enrollment trends and responses from our survey indicate a necessity to have continuity to this subject matter and offer other lower level classes in gaming and join efforts and resources with other programs such as Multimedia, Art programs to develop a gaming certificate.

Curriculum

PCC Core Outcomes for Computer Science

College Core Outcome Mapping	
PCC Core Outcomes	CS Core Outcomes
Communication Communicate effectively by determining the purpose, audience and context of communication, and respond to feedback to improve clarity, coherence and effectiveness in workplace, community and academic pursuits.	Computer Science is a social activity where problem solving occurs in teams, and communication of technical information is an ongoing requirement. CS courses address this through team projects and class presentations, verbal and written reports, and the analysis phase of the systems development life cycle.
Community & Environmental Responsibility Apply scientific, cultural and political perspectives to natural and social systems and use an understanding of social change and social action to address the consequences of local and global human activity.	The social, ethical, and environmental impact of technology is addressed in several courses in the CS program, particularly in CS 160. Course material on the ethical use of information technology, privacy, intellectual property, and the impact of technology on the natural environment is presented and discussed.
Critical Thinking and Problem Solving Identify and investigate problems, evaluate information and its sources, and use appropriate methods of reasoning to develop creative and practical solutions to personal, professional and community issues.	Critical thinking and analysis is at the core of Computer Science. Students learn systematic problem analysis techniques applicable to information technology, including a variety of software design methodologies. Abstract reasoning is emphasized in the mathematics and object-oriented topic areas of the curriculum.
Cultural Awareness Use an understanding of the variations in human culture, perspectives and forms of expression to constructively address issues that arise out of cultural differences in the workplace and community.	The multi-cultural aspects of computer technology are emphasized horizontally by discussion of topics such as the digital divide in CS 160, and vertically by discussion of the human-computer interface and interface design in CS 161, and CS 162. Students are introduced to the different ways cultures perceive information and technology, and become familiar with the variety of technologies and practices available for manipulating and communicating information. Technologies for supporting human disabilities are discussed as well as technologies for supporting diverse social cultures. Local culture can influence how

	suppliers communicate, make decisions, complete tasks and resolve conflicts. Such differences can be especially problematic in multi-sourcing deals, which may involve suppliers from around the globe.
Professional Competence Demonstrate and apply the knowledge, skills and attitudes necessary to enter and succeed in a defined profession or advanced academic program	The focus of the Computer Science program is the development of competencies needed for success in four-year CS programs. Students master rigorous academic skills rather than professional or vocational skills.
Self-Reflection Assess, examine and reflect on one's own academic skill, professional competence and personal beliefs and how these impact others.	All of the CS courses involve rigorous assessment of the student's technical proficiency and learning skills, and students incorporate this feedback to improve their expertise. The CS 160 course includes lessons on the social and ethical issues surrounding computer technology. The CS 160 and CS 133G courses require students to explore and describe their personal preferences and beliefs regarding specific types of computer technology.

Program Support for PCC Core Outcomes

The table below shows how Computer Science courses map to the PCC Core Outcomes, which each row of the table representing a course and the five rightmost columns representing the Core Outcomes. The labels for the PCC Core Outcomes are:

1. **C01** – Communication
2. **C02** – Community and Environmental Responsibility
3. **C03** – Critical Thinking and Problem Solving
4. **C04** – Cultural Awareness
5. **C05** – Professional Competence
6. **C06** – Self-Reflection

The numeric mapping level indicators are:

1. Not Applicable.
2. Limited demonstration or application of knowledge and skills.
3. Basic demonstration and application of knowledge and skills.
4. Demonstrated comprehension and is able to apply essential knowledge and skills.
5. Demonstrates thorough, effective and/or sophisticated application of knowledge and skills.

Course #	Course Name	CO1	CO2	CO3	CO4	CO5	CO6
CS133U	Introduction to C	3	2	4	1	3	1
CS133G	Introduction to Computer Games	3	2	4	3	3	3
CS140U	Introduction to UNIX	3	2	4	2	4	1
CS160	Exploring Computer Science	3	3	4	3	3	3
CS161	Computer Science I	3	2	4	1	4	2
CS162	Computer Science II	3	2	4	1	4	2
CS201	Computer Systems	3	2	5	1	4	1
CS 233G	Game Programming	3	3	5	1	5	1
CS 233U	Advanced C Programming	3	3	5	1	5	1
CS 250	Discrete Structures I	3	2	5	1	5	1
CS251	Discrete Structures II	3	2	5	1	5	1
CS 260	Data Structures	3	1	5	1	5	1
CS261	Programming Systems	3	1	5	1	5	1

Program Support for ACM Outcomes

The Association for Computing Machinery[1] (ACM) has developed a model curriculum for two-year colleges that serves as a general standard for PCC's Computer Science program[2]. PCC utilizes what the ACM terms a 'breadth-first' approach with CS 160 as the entry course in the program. This approach was strongly recommended by ACM in the 2001 report, with some improvements in the 2008 report.

> "According to the American Association for Community Colleges, nearly one-half of all undergraduate students in the United States are enrolled in two-year colleges. One segment of the two-year college student population begins their academic careers intending to transfer to a university in order to continue their academic studies. These students must be served by well-defined articulation agreements, programs of study consistent with those found in university settings, and faculty sensitive to the special issues associated with such students and programs."

The Computer Science program at PCC is a party to transfer guides with PSU, OSU, UO, and OIT. Students who successfully complete their first two years of study in Computer Science at PCC can then transfer seamlessly to these schools for their third and fourth years. A detailed analysis of how the CS program at PCC maps to the lower division programs at various Oregon four-year institutions is presented in Appendix 4. This shows that our program is very well designed to meet the requirements of students who wish to transfer to one of these institutions with junior level standing.

The Program incorporates 14 of the ACM recommended 63 core topics as noted in chart. Key to topics noted below chart.

Course	ACM Body of Knowledge Core Topic
CS 133U	PF/Fundamental Constructs PF/Algorithmic Problem Solving PF/ Data Structures PL/ Declarations and types
CS 133G	PF/Fundamental Constructs PF/Algorithmic Problem Solving AL/Basic Analysis PL/Object Oriented Programming HC/Building GUI Interfaces HC/User Centered Software Evaluation GV/Fundamental Techniques GV/Computer Animation GV/Virtual Reality SP/History Of Computing SP/Social Context SP/Professional Ethics SP/Intellectual Property SP/Privacy And Civil Liberties
CS 140U	OS/Overview of operating systems OS/Operating system principles OS/ Concurrency
CS 160	DS/ Functions, Relations, and Sets AL/ Basic Computability AR/ Machine level representation of data

	OS/ Overview of operating systems PL/Overview of programming languages PL/Virtual Machines PL/ Introduction to language translation HC/ Foundations of human-computer interaction IS/Fundamental issues in intelligent systems SP/ Social Context of Computing SP/ Professional and ethical responsibilities SP/ Intellectual property SP/ Privacy and civil liberties
CS 161	PF/Fundamental Programming Constructs PF/ Algorithms and Problem-Solving PF/ Fundamental Data Structures PL/ Declarations and types PL/ Abstraction mechanisms PL/ Object-oriented programming HC/ Building a simple graphical user interface SP6 Intellectual property
CS 162	PF/Recursion PF/ Event-Driven Programming AL/ Basic Algorithmic Analysis AL/ Algorithmic Strategies AL/ Fundamental Computing Algorithms PL/ Abstraction mechanisms PL/ Object-oriented programming GV/ Fundamental techniques in graphics SE/Software Design SE/ Using APIs SE/ Software tools and environments
CS 201	AR/ Digital Logic and Data Representation AR/ Computer Architecture and Organization AR/ Machine level representation of data AR/ Assembly level machine organization AR/ Memory system organization and architecture AR/ Interfacing and communication AR/ Functional organization
CS 234U	AL/Fundamental Computing Algorithms SE/Using APIs SE/ Software tools and environments
CS 260	AL/ Fundamental Computing Algorithms PF/ Algorithms and problem solving PF/ Fundamental data structures PF/ Recursion PF/ Event-driven programming
CS 261	AR2 Machine level representation of data OS3 Concurrency OS4 Scheduling and dispatch OS5 Memory management
CS 250	DS/ Functions, Relations, and Sets DS/ Basic Logic DS/ Proof Techniques DS/ Basics of Counting
CS 251	DS/ Graphs and Trees DS/ Discrete Probability

Key: Fourteen (14) Knowledge Areas

- Discrete Structures (DS)
- Human-Computer Interaction (HC)
- Programming Fundamentals (PF)
- Graphics and Visual Computing (GV)
- Algorithms and Complexity (AL)
- Intelligent Systems (IS)
- Architecture and Organization (AR)
- Information Management (IM)
- Operating Systems (OS)
- Social and Professional Issues (SP)
- Net-Centric Computing (NC)
- Software Engineering (SE)
- Programming Languages (PL)
- Computational Science (CN)

Course Outcomes and Assessment Strategies

Outcomes

Computer Science courses include a wide variety of class outcomes. The CS 162 Computer Science II and CS 201 Computer Systems outcomes shown below are examples of typical class outcomes in the Program. These outcomes are designed to reflect PCC's Core Outcomes as well as transfer requirements.

CS 162 - Computer Science II

On completion of this course students should be able to:

- Employ a deep knowledge of the procedural paradigm and a recognized software development methodology to develop computer programs.
- Design and construct simple object-oriented software with an appreciation for data abstraction and information hiding.
- Effectively use software development tools including libraries, compilers, editors, linkers and debuggers to write and troubleshoot programs.
- Be successful in subsequent college level Computer Science coursework.

CS 201 - Computer Systems

On completion of this course, students should be able to:

- Employ a deep knowledge of computer systems architecture to enhance the correctness, performance, and utility of the application or system software that they may write.
- Identify how such programs work and how to troubleshoot and fix them when they break.
- Be successful in subsequent college level Computer Science coursework.

Assessments

The CS program has synchronized course outcomes and assessments, and revised several CCOGs this past year. The list of assessment tools below address the spectrum of class outcomes in the Program:

- Service Learning
- Contextual written tasks in or outside of class.
- Individual projects.
- Presentations
- Quizzes and/or examinations.
- In-class interactive role-plays
- Participation and Group Discussion
- Self-Assessment

The primary goal of the Computer Science program at PCC is to prepare students for transfer to four-year Computer Science programs. The assessment strategies are more detailed than the intended

outcomes and most of them address one or more outcomes. An example of this is the assessment list for CS 162:

CS 162 Computer Science II

Students will complete software project(s) comprised of object-oriented designs, implementations, and test plans.

- Designs will demonstrate the use of good object-oriented design principles.
- Implementations will include C++ stream input/output and file processing.
- Test plans will include test cases demonstrating both black and glass box testing strategies.

Students will demonstrate competency on exam(s), which are constructed to demonstrate:

- Written technical communication and effective use of terminology
- Facility with programming language syntax and semantics
- Ability to read and understand specifications, designs and programs
- Application of concepts
- Individual capability in design, implementation and testing of program components

Grade Distribution

The Program monitors the grade distribution by course and by campus as a quality assurance technique. A significant difference among classes in a sequence, or offerings among campuses, may indicate a curriculum or teaching issue that SAC should address. The table below includes the most recent available grade data, and while some differences are apparent the faculty feels the data shows, the Program is reasonably consistent in assessing student achievement.

Grade distribution for CS Credit Courses for 2009-10 by Campus and Course

		A	B	C	D	P	F/NP	W	Inc/Audit	Total
CS 133G	Sylvania	39.6	11.6	10.1	2.9	0.5	17.4	16.4	1.4	207
	Rock Creek	20.6	20.6	26.5	17.6		2.9	11.8		34
CS 133U	Sylvania	31	16.1	8	2.9	0.6	20.7	16.7	4	174
	Rock Creek	33.3	15.6	4.4	2.2		24.4	20		45
CS 140U	Sylvania	42.6	22.1	7.4	5.9		8.8	11.8	1.5	68
	Rock Creek	37	14.4	12.3	8.2		22.6	4.8	0.7	146
CS 160	Sylvania	47.2	20.6	9.4	3.9		6.1	12.8		180
	Rock Creek	39.7	20.6	13.5	7.1		15.6	2.8	0.7	141
CS 161	Sylvania	36.7	16.7	8.8	6.7		12.5	15.8	2.9	240
	Rock Creek	38.3	16.8	12	4.8		19.2	7.2	1.8	167
CS 162	Sylvania	43.6	15.5	9.1	2.7		12.7	16.4		110
	Rock Creek	38.2	28.9	5.3		3.9	13.2	10.5		76
CS 201	Sylvania	75.9	17.2				3.4	3.4		29
	Rock Creek	38.5	30.8	15.4			11.5		3.8	26
CS 233G	Sylvania	78.9	5.3					15.8		19

CS 233U	Sylvania	60				20			20	5
CS 260	Sylvania	38.8	30.6		2		20.4	6.1	2	49
	Rock Creek	44.4	22.2				25	8.3		36
CS 261	Sylvania	65	10					20	5	20
	Rock Creek	45	30	10				10	5	20

Course Content and Outcome Changes

Completed Changes

- The CS SAC discussed and voted to reactivate CS250 and CS251 (Discrete Structures I and II) instead of MTH231 and MTH232 to best meet the needs of our CS students who wish to transfer to PSU. After an email discussion between Colin Goble (Dept Chair, SY), and Professor Warren Harrison of PSU, it was decided that it would be best to teach these courses within the CS department.

- CS 200 (Computer Systems I) has been deactivated since OSU and PSU do not offer those courses anymore. We will offer CS201, as Computer Systems and a portion of the content from CS200 (Computer Systems I) will be covered in CS201 (Computer Systems).

- PSU has also revised CS161 (Computer Science I) and CS162 (Computer Science II) to help transition students from other programming languages like Java, Gaming software etc. CS161 (Computer Science I) is no longer a recommended course for those taking CS162 (Computer Science II), but they will have to pass a test to prove they have the basics of programming. The PCC CS SAC has decided to keep CS161 (Computer Science I) and CS162 (Computer Science II) in C++ and make some minor changes to content and description.

- The CS Department has introduced CS233U (Advanced Programming in C) to keep abreast with technology. Doug Jones, Sylvania instructor, has proposed and is prepared to teach iPhone applications in this class. This will require a Mac lab with about 25 workstations.

- We have added CeeBot to our CS 160 (Exploring Computer Science), which has been enthusiastically received by our students. CeeBot relates to all the concepts of Computer Science such as algorithms, Programming, Objects and it is fun for the students. We have incorporated online material and there is no need for a textbook.

- CS 233U (Advanced Programming in C) has incorporated iPhone applications in his class. It fits into the course, is fun for the students, promotes communication between them, reinforces computer concepts, and is a highlight for them to remember the course.

- CS 161 (Computer Science I) and CS 162 (Computer Science II), Colin Goble and Michael Trigoboff have added a component on graphics programming and game assignments to make these classes more relevant and exciting. The technology being used is Dark GDK.

Collaborative Content Changes

CS 233G, the advanced gaming class has been offered in collaboration with OIT. The class has been offered jointly by PCC and OIT and has been outstandingly successful.

Pending Changes

An examination of the CCOGs found that some courses need outcome revisions regarding their descriptions. Outcomes should express what the student would be able to do in a job setting. Outcomes using verbs such as “explain” or “describe” are unacceptable within a job setting. The courses listed below need some of their student outcome revised.

- CS 133U (Introduction to C)
- CS 140U (Introduction to UNIX)
- CS 161 (Computer Science I)
- CS 233U (Advanced Programming in C)
- CS 260 (Data Structures)
- CS 261 (Programming Systems)

These revisions will be implemented when the course CCOGs are revised for their 3-year approvals.

---

[1] ACM is a member society of CSAB, along with the Institute of Electrical and Electronics Engineers – Computer Society (IEEE-CS) and the Association for Information Systems (AIS). CSAB is the lead society within ABET for accreditation of programs in computer science, information systems, and software engineering, and is a cooperating society for accreditation of computer engineering.

[2] Computing Curricula 2008 Computer Science Final Report. The Joint Task Force on Computing Curricula IEEE Computer Society/Association for Computing Machinery © 2001 ACM/IEEE

Distance Learning

CS has been offering Distance Learning (DL) courses since 2002. All active CS classes, except for those listed below, are offered in the DL modality.

- CS233U Advanced C Programming
- CS233G Advanced Gaming
- CS250 Discrete Structures I
- CS251 Discrete Structures II

The following revelations and concerns regarding Distance Learning are highlighted:

Revelations

- Students are getting proficient with technology and Blackboard. More and more instructors are using blackboard for hybrid classes, and this has helped students. Students are more willing to participate in discussion and are able to take online classes confidently.
- Students taking online CS classes get the added benefit of being able to download and install their own software. They must configure and troubleshoot their own software and this is an added benefit for students in this field.
- Most of the online classes in our department have been modeled similarly and this may benefit students.
- Most of our instructors blend the on campus class with the online class resulting in a hybrid class, which helps improve PCC's green efforts.

Concerns

- Blackboard is not consistent and is sometimes unstable. The main issue seems to be when students are taking their quizzes. Blackboard times out when they are in the middle of a quiz, and by the time they log back in, their quiz time is over.
- Plagiarism on the Internet is a big concern for online classes. It is quite hard to tell if the students are cheating, or if they are submitting their own work.
- Tutoring for CS students is available only in the SY campus. It would be beneficial to have tutors at the Rock Creek Campus. It would also be helpful to have online tutoring for the online students.

Needs of Students and Community

A Look at Computer Science Students

Nationwide enrollment in Computer Science programs is increasing (Taulbee, 2009). PCC enrollment in Computer Science courses at the end of the 2009 years stood at 246 FTE (up 15% from the previous year), and 2010 enrollment is up 34% compared with 2009 (through week 4). Informally, most on-campus CS classes are full and many classes have waiting lists.

Demographic data shows that most CS students are "white, non-Hispanic" (76.9%), followed by "Asian/Pacific Islander" (12.9%). Approximately 17% of the CS students are female, half are full-time students, and 73% are 30 years of age or younger.

Most (83%) are degree-seeking students. This data is consistent with National and Historical Profiles of Computer Science students.

Enrollment trends and demographic data suggest a continuing need to expand and upgrade Computer Science facilities. Beyond the ability to enroll in their chosen courses, students need access to support resources such as computer labs and subject matter tutors to facilitate their progress in the program.

Persistent Needs

There are a number of student needs that are unchanged from earlier reviews:

1. CS students need to understand the scope and nature of the discipline in order to assess their level of interest, their aptitude, and their academic or professional goals

2. Students need unambiguous information about academic and career paths in Computer Science, including transfer requirements, degree requirements, and career opportunities.

3. Students need to develop the academic and technical skills needed for university matriculation

4. Students need access to educational resources, and need the ability to utilize educational resources in a manner compatible with their individual requirements

Each of these persistent needs involve a dynamic that requires PCC to adapt new methods of meeting student needs. In the first case, according to the Association for Computing Machinery (ACM 2008), the core content of the CS discipline should now include:

- Emphasis on information security, particularly the ability to write safe and secure code

- Increased exposure to concurrent processing design and practice
- Awareness of net-centric computing practices

- Greater awareness of software engineering principles and the requirements of complex systems development

Meeting the CS student's need for a clear understanding of the scope of the discipline requires PCC to incorporate these elements into our survey courses (e.g. CS 160) as well as into the structure of our programming assignments.

The need for accurate transfer and career information requires PCC to have an ongoing process of articulation review, and periodically to solicit new partner institutions. For example, Portland State University recently revised their CS transfer requirements, resulting in corresponding adjustments in PCC's CS program.

The need for students to access educational resources in a manner that fits their individual situations drives much of the innovation in class delivery at PCC. This has resulted in increased distance learning opportunities, elimination of some textbook requirements, and modifications to course scheduling practices. Additional student support services, such as increased online tutoring, virtual team support, and real-time distance learning interactions among students and faculty may be required to meet this ongoing need.

Community Needs

At least three larger communities include the CS program, and each has its own set of stakeholders and stakeholder needs. The most immediate and influential community is the College itself. The CS program must serve the needs of PCC, which include:

- Support for College-wide goals and outcomes
- Adequate enrollment and appropriate utilization of PCC resources
- Appropriately skilled faculty and staff available to support student needs

The needs of the College community have evolved since 2002, and two that relate directly to the CS program are:

- Assessment of the core outcomes are more strongly emphasized in the accreditation process, with special emphasis on critical thinking skills during this academic year
- Sustainability and environmental responsibility have become part of the College's community mission, and several initiatives related to sustainable technology (e.g. electronic recycling and carbon-neutral electrical generation) are active in the community

The need to assess core outcomes and ensure that CS program outcomes align with College outcomes influences how the CS program evaluates student learning. As well as, how course outcomes are identified and articulated. An awareness of sustainability issues is an important component in discussing the ethical use of technology (a topic included in several CS courses).

Our partner institutions, notably Portland State University and Oregon Institute of Technology, form a second community to which the CS program must answer. PCC maintains a network of formal and informal transfer agreements, and these constitute a commitment by the CS program to ensure that transfer students have the opportunity to learn a specific set of academic skills that will enable them to succeed in their upper division courses. The CS program avoids the role of "gatekeeper" for transfer institutions, and balances this community's need for adequately prepared students with the student's need for access to educational programs. PCC never prohibits students from transferring, nor does PCC ever strive to "weed out the weak" students. Instructors in the CS program actively encourage all

students to pursue their educational goals while simultaneously attempting to provide a realistic assessment of their skills and aptitude in the discipline.

The need of partner institutions has changed since the last review. Both PSU and OIT have changed their CS transfer requirements (in response to their stakeholder's needs) and this has driven a number of changes in the CS program at PCC. Some of the changes include:

- CS 161 is no longer a required transfer course at PSU. Students may enter CS 162 if they have one term of any high-level programming language

- Students transferring to PSU may substitute any laboratory science for the Physics transfer requirement. CS students may take Biology or Chemistry, for example, instead of Physics.

- The Information Systems degree at PSU has been discontinued due to accreditation issues. CS students at PCC are no longer able to transfer to the ISQA program

- Computer game development is a new option in some partner institutions, e.g. the Art Institute of Portland. The CS program at PCC has developed three game-related courses (CS 133G, CS 233G, and an edition of CS 233U) to support this interest, and several PCC programs have discussed creating an interdisciplinary 2yr degree in computer game development.

This is by no means an exhaustive list of changes in the partner community, but is meant to illustrate the continual process of changes that occur. The CS program makes some type of course change virtually every term to support partner needs, and this is expected to continue. In addition, our program must also cater to the needs of high school students and students pursuing Engineering degrees.

A third community that influences the CS program is the high technology industry. The industry employs many PCC alumni, and the concerns of local high technology firms drive many of the needs of our partner institutions as well as our own service offerings. In a broader sense, the CS discipline is driven in large part by the needs of the high technology industry.

Industry is concerned that CS graduates lack essential industry skills (Stroustrup, 2010). Academic and industry organizations have responded to these concerns by revising recommended curricula (e.g. ACM 2008, Stroustrup, 2010), by revising recommended pedagogical methodology (e.g., Felleisen, 2009, Stephenson, 2009), and by recommending new types of job roles in industry (e.g. Coder, 2009).

There is a developing consensus among academic and industry observers that high technology is transitioning from the "information age" (1990 to 2001 or so) to the current "conceptual age" (Zyda 2009). A strong emphasis on creative and cognitive processes such as design, artistry, and empathy characterizes conceptual age industry needs. Traditional software programming and engineering skills are required, but are no longer sufficient for industry success.

Conceptual age requirements manifests in CS programs primarily in two ways. The first is an emphasis on collaborative design and implementation, with a focus on team skills and learning to build highly functional teams. PCC's partner institutions meet this need in their upper division courses, but this will eventually become a lower-division requirement as well.

The interest in game development as part of the CS curricula is a second manifestation of conceptual age industry requirements. Game development emphasizes cognitive skills over implementation skills: students may develop games using engines (e.g. Game Maker in CS 133G) before learning the engineering skills to develop games in a traditional programming language such as Java or C#. Industry needs a workforce with these cognitive skills in addition to strong engineering skills, and game development is beneficial training for jobs unrelated to creating computer games.

The requirement for training conceptual age cognitive and creative skills appears to be long-term industry need, and this in turn is likely to spur pedagogical development. Game development is more effective than traditional programming in developing these skills, but the development of more effective methods is ongoing in many university-level CS programs.

Faculty

There are three full-time faculty at Sylvania and two full-time faculty at Rock Creek teaching in the Program, and there are at least four part-time faculty teaching in the Program each term.

The Computer Science department has successfully recruited one new full-time female instructor for the Rock Creek campus in the last three years. In this case, it was a replacement for faculty who resigned. All instructors have Master's degrees, and most have substantial industry experience and/or prior teaching background in Computer Science at the college level. The overall level of experience and teaching expertise in the department has been maintained or enhanced by our hiring practices.

The diversity mix of Computer Science instructors and its students is a good example of PCC diversity goals. For example, our female instructors show leadership in the field in for a program where female students do not dominate it. In addition, internationalization efforts bring international perspectives to our program.

The adjunct body of instructors is well educated and currently active in their fields of expertise. They are placed to teach courses where their expertise is valuable to our students about to transfer out of our program, offering precious advising regarding the industry that they are involved in. Their current working background includes Intel, Tektronix and other software companies in the area. Many of these instructors have been in the program for over 10 years; a few are active participants of our SAC meetings providing valuable input. Many are DL certified allowing flexibility for the courses that they can teach, ready to fill in, when needed.

College-wide PT/FT Ratio

Percent of Sections Taught by Full-Time Faculty

College Total (SY, CA, RC, ELC)	Fall 05	Fall 06	Fall 07	Fall 08	Fall 09
Total Sections	2,843	2,832	2,973	3,206	**3,535**
Sections taught by ft faculty	1,245	1,223	1,275	1,291	**1,274**
Sections Taught by FT Faculty (%)	43.8%	43.2%	42.9%	40.3%	**36.0%**

Computer Science PT/FT Ratio

Percent of Sections Taught by Full-Time Faculty

College Total (CA, RC)					Winter 10
Total Sections					**28**
Sections taught by ft faculty					**15**
Sections Taught by FT Faculty (%)					**52.0%**

Instructor Qualifications

Current instructor qualification follows:

- CS Instructor
- Education: Master's Degree in Computer Science or Software Engineering is preferred.
- Minimum requirements shall be a Masters degree in a related field such as Electrical Engineering, Math, Computer Engineering, Information Systems or Systems Science. Preference will be given to candidates with relevant recent industry or education/teaching experience.

CS Faculty Professional Development Activities

The Computer Science faculty has been engaged in the past years in a variety of activities to support our program. Communication with members of the software community and faculty from partners universities keep us informed of upcoming events and curricular changes that most faculty are engage and participate in. Such events consist of state wide department chair meetings, faculty visits to other institutions, participation in IT shows, speeches at local High Schools, visits to foreign universities to support PCC's Internationalization efforts, preview days, advising days, among others.

Other activities include:

- Attending online courses, "Teaching for Learning", and "Succeeding with online group projects", and participation in the TIP (Teacher Improvement Program) to deliver content effectively to students.
- Studying the world of computer gaming, the technologies which drive it, and the potential utilization of those technologies to computer science education with specific focus on making the program at PCC more relevant, interesting, and challenging to our student base.
- Attended several conferences and interest groups that have had a significant impact on the computer science program as a result. We have introduced robotics and 2D game development at an introductory level, 3D game programming at a more advanced level, and introduced several examples from the world of game programming throughout our programming sequence of classes.
- Participating in group projects of providing a "computer games" aspect to programming assignments. Evaluation of a number of different gaming environments, and participated in the decision to work with Dark GDK/Visual C++.
- CS adjunct faculty also brings experience and expertise to the group: responsibility for engineering teams responsible for several RF and Microwave Instrument Products relevant to the PCC Computer Science program.
- Training on Agile Software development practices, and deployment Scrum methodology for two programming teams. Introduction of computer micro architectural topics in courses, including but not limited to high performance techniques used in the current generation of Personal Computers.
- Development of Advanced C course learned Objective C, learned to write Cocoa applications on OS X and iPhone OS 3, and written several iPhone applications and games. Learned openGL and

darkGDK so that games and graphic applications can be written in C/C++ on Linux and Windows systems for our C++ programming classes.

- Design and implementation of the online CS162 class to align with the curriculum change at PSU. With this new edition, 20+ videos were recorded to make the online class more student friendly.
- PCC's Distance Learning training sessions.
- SAMBA server training.
- Internationalization Committee activities.
- Career fair participation.
- TLC workshops.

Facilities and Support

1. One or more computer labs (Windows or Mac) will help with the current classroom scheduling and provide greater computer access to our students. It will also provide the hardware foundation for introducing more new exciting classes to enrich our program.
2. A tutoring facility for Rock Creek campus is needed to serve our students in that area.

Professional Development Support:

CS instructors are constantly updating our knowledge and skills to stay current with the rapidly changing high tech industry. The basic support needed include:

- Purchasing technical books and software
- Self-teaching is still the most economical way for instructors to pick up new technology, but it is hard to do it without a good book or the required software.
- Attending conferences and seminar
- Bringing back information and new ideas from professionals at other colleges or industry.
- Hearing what works and what does not work from people who have had the practice and apply it to our own classes or curriculum.
- Participating in conference to get the CS program's name out and make it more visible.
- Networking with instructors from other colleges and engineers from industry makes future information/best practice sharing easier and fun.
- Taking classes and industry standard exams
- Learning from experts is a faster way to increase our skill set. A classroom environment presents opportunities to interact with fellow students. It is also beneficial to experience being a student again after teaching many years so that we can understand our students better.
- Being certified from the industry standard exams: we would certainly be pushed to master a technology within a timeframe. We cannot only share our knowledge gained from the exam preparation, but also the experience of actually taking it with our students who might need the knowledge in the future.

Industry connection support

- Bringing in guest speakers from local high tech industry
    - This allows our students to meet people who are leading in the industry. If the students can see themselves in the speakers' role in 5 or 10 years, they will be more motivated to complete the program and continue with their education. The conversation could make their goal more tangible and the impact could go well beyond our expectation.
- Sending students out for internships

  Let the students meet the world. They can apply what they learn in the classroom and gain working experience and new perspectives. Most likely, when the students come back to the classroom, they appreciate the learning opportunity more. They can share their experience with their classmates and inspire more students to try.

Recommendations for Improvement

The Computer Science program must continue to evolve to meet the needs of our students, stakeholders, and larger communities. The recommendations presented below guide this evolution.

1. *Work with the College leadership and the College community to expand the instructional technology resources available to Computer Science students*

   Computer Science is one of many technology-intensive programs supported by the College, and expansion of technology resources is a College-wide issue. The technology issues of interest to the Computer Science program include:

   a. Increasing the capacity of our computer-equipped classrooms. Class size is limited by the number of workstations available in a room, and sections are limited by the number of computer-equipped classrooms

   b. Increase the diversity of computer equipment in the classrooms to include Mac workstations, game consoles, and other technology specific to Computer Science courses

   c. Increase the availability of mobile devices for classroom instruction so that courses can include, for example, demonstrations of Android or iPhone software, multi-touch user interfaces, and other technologies specific to mobile devices

2. *Innovate class delivery and pedagogy to reduce student costs*

   Reducing student cost increases student access to the Computer Science program. Many student expenses are beyond the scope of the Program, but a number of cost reductions are possible, including:

   a. Eliminating hardcopy textbooks in favor of electronic or online reference material

   b. Using electronic distribution and submission of graded assignments and assessments to reduce paper use in the on-campus courses

   c. Continuing to innovate on-campus course scheduling procedures to reduce conflicts with student work schedules, reduce travel and parking times, and improve the utilization of College resources

3. *Increase faculty expertise in the discipline of Computer Science and the teaching of Computer Science topics*

   The College has long recognized and supported the need for ongoing training of faculty, and Computer Science faculty expect to better utilize College resources to maintain their high level of expertise in the discipline with a focus to:

   a. Increase conference attendance to heighten awareness of current developments in the discipline. College resources available to support conference attendance are under-utilized by Program faculty

b. Increase knowledge of current literature. College resources are available for journal and magazine subscriptions

4. *Continue to implement innovative curriculum changes*

   The core content of the Computer Science curriculum is determined by the needs of our transfer partners, but the Program has always included additional outcomes that reflect College or industry priorities. The Program can better reflect these priorities by integrating the following in our curriculum:

   a. Increase emphasis on collaborative skills needed for effective teamwork
   b. Increase emphasis on cognitive and creative skills to analyze problems and design solutions
   c. Increase emphasis on engineering skills, for example
      i. Project management
      ii. Large program development (multi-source code)
      iii. Source code and version control management
      iv. Problem identification and debugging

5. *Work with College leadership and the College community to expand outreach efforts to K-12 and industry stakeholders*

   The College has expanded and improved its marketing and outreach capability since the last program review, and the Program seeks to utilize this expertise to improve outreach to K-12 students (particularly High School students) and employees in the local high-technology industry. In particular, we will

   a. Develop an outreach program for local K-12 students, particularly High School students, to ensure students are aware of the benefits of enrolling in Computer Science at PCC

   b. Develop an outreach program for local technology employees to support their academic advancement. Many of our existing students are employees seeking specific job skills or a degree that will improve their workplace value.

Appendix A – Employment Trends

This Appendix contains **national** level data for Computer Science related employment.

Employment Projections from the National Employment Matrix							
Occupational Title	SOC Code	Employment, 2008	Projected Employment, 2018	Change, 2008-18		Detailed Statistics	
				Number	Percent		
Computer and information scientists, research	15-1011	28,900	35,900	7,000	24	[PDF]	[XLS]
NOTE: Data in this table are rounded. See the discussion of the employment projection table in the *Handbook* introductory chapter on *Occupational Information Included in the Handbook*.							

According to Bureau of Labor Statistics, "Computer software engineers are among the occupations projected to grow the fastest and add the most new jobs over the 2008-18 decade, resulting in excellent job prospects."

Employment of computer software engineers and computer programmers are projected to increase by 21% by 2018, much faster than all other occupations. Expanding Internet technologies have spurred demand for computer software engineers who can develop Internet, intranet, and World Wide Web applications. Likewise, electronic data-processing systems in business, telecommunications, healthcare, government, and other settings continue to become more sophisticated and complex.

According to MS NBC media, Computer Science holds the fourth position as the highest paid Bachelor's degree in 2010, with an average starting salary of $61, 205 (http://msnbcmedia.msn.com/i/CNBC/Sections/News_And_Analysis/_News/_SLIDESHOWS/HighestPaidDegrees/SS_highest_const_comp.jpg).

Projections by National Employment Matrix				
Occupational Title	Employment, 2008	Projected Employment, 2018	Change, 2008-18	
			Number	Percent
Computer software engineers and computer programmers	1,336,300	1,619,300	283,000	21
Computer programmers	426,700	414,400	-12,300	-3
Computer software engineers	909,600	1,204,800	295,200	32
Computer software engineers, applications	514,800	689,900	175,100	34
Computer software engineers, systems software	394,800	515,000	120,200	30
NOTE: Data in this table are rounded. See the discussion of the employment projection table in the *Handbook* introductory				

Projections by National Employment Matrix				
Occupational Title	Employment, 2008	Projected Employment, 2018	Change, 2008-18	
			Number	Percent
chapter on *Occupational Information Included in the Handbook*.				

According to the Veritude 2009 IT Outlook Report, "Programming Language skills (C, C++, C#) were in low demand in Q2 of 2008; by end of the year, however, they have emerged as a highly desired skill set, being cited as "in demand" just as often as Business Intelligence and Enterprise Solutions skills."

Employment Forecast

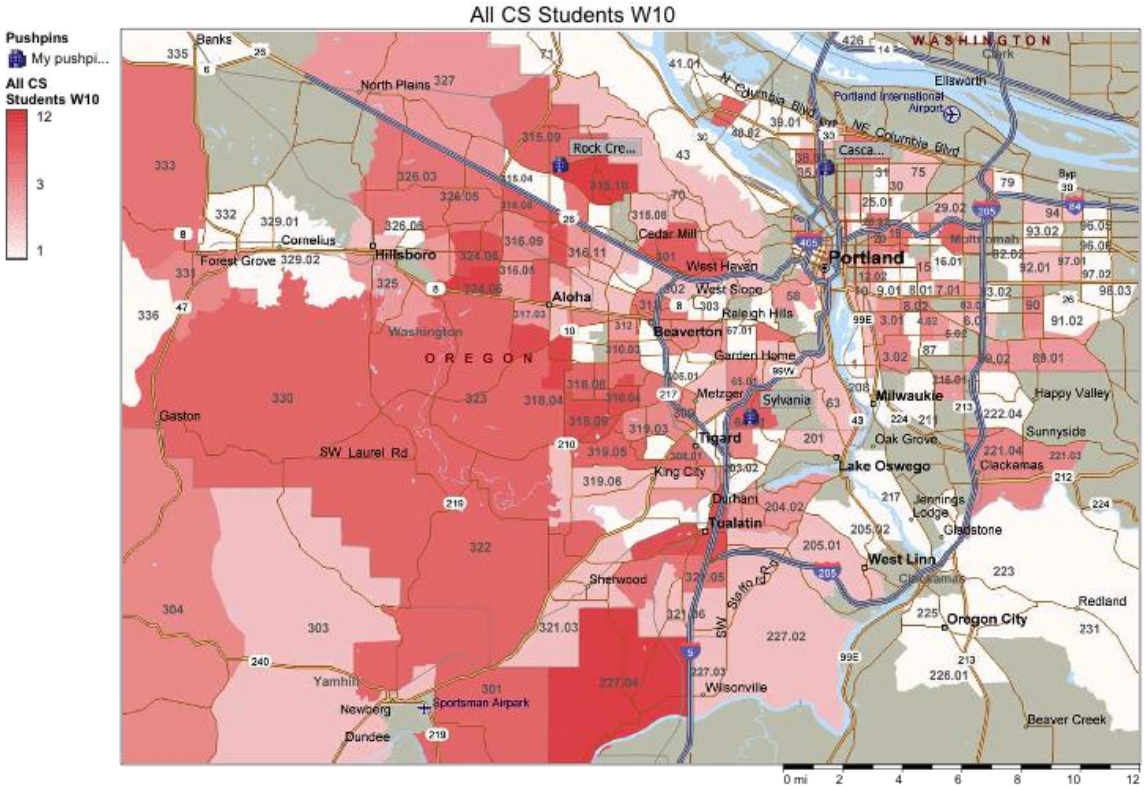
The following table from The Riley Guide details jobs, salaries, employment count and potential growth in the Computer and Software field. Data taken from: http://www.rileyguide.com/careers/index.shtml#comp.

Computer and Software Earnings

Career	Degree Holders	Avg. Wage	Employee Count	Growth in Jobs 2006-2016
Computer Scientists	68%	$97,970	26,610	22%
Computer Specialists	68%	$75,150	191,780	15%
Computer Support Technicians	43%	$43,450	545,520	13%
Computer Systems Analysts	68%	$75,500	489,890	29%
Data Network Administrators	50%	$66,310	327,850	27%
Database Administrators	72%	$69,740	115,770	29%
Networking Analysts	57%	$71,100	230,410	53%
Software Application Engineers	85%	$85,430	494,160	45%
Software Programmers	73%	$69,620	394,230	-3%
System SW Engineers	85%	$92,430	381,830	28%

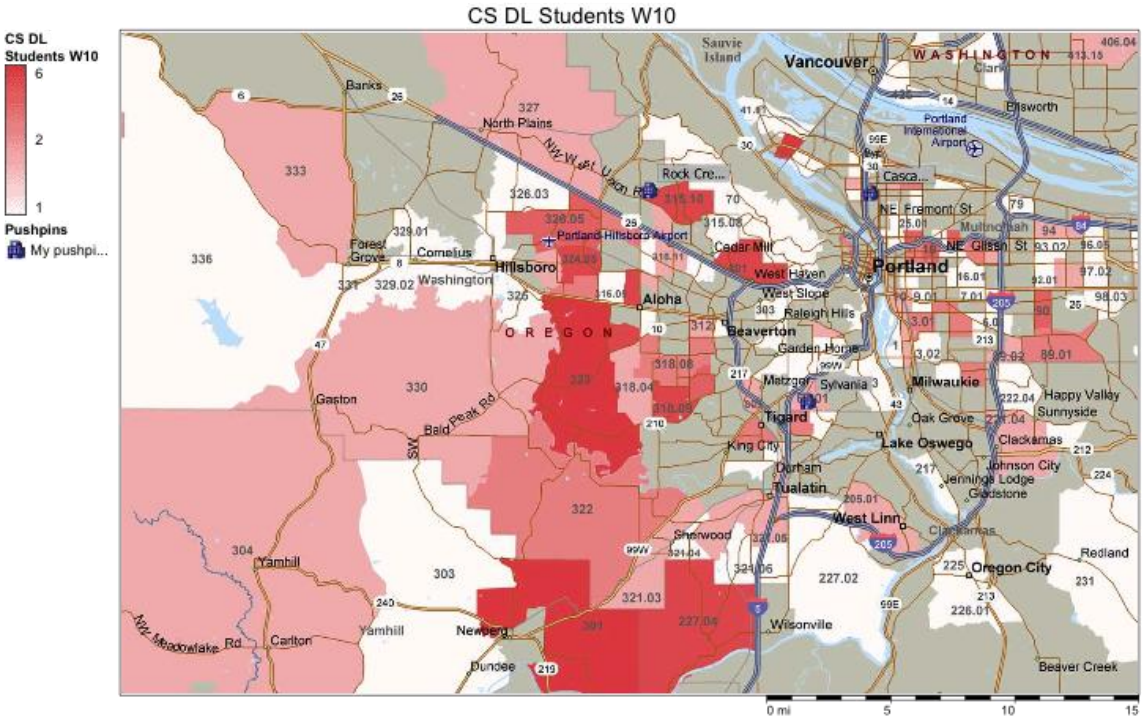
Appendix B - Winter 2010 CS Students

The map below shows the geographic distribution of Winter 2010 Computer Science students.



Appendix C - Winter 2010 CS DL Students

The map below show the geographic distribution of Winter 2010 Computer Science Distance Learning students.



Appendix D – University Transfer Guides

Portland State University

Portland Community College
Transfer Center
2009-2010

Transfer Guide for:
Portland State University
Computer Science Major

Lower Division Department Requirements for the Computer Science Major:

PSU Courses	Quarter Credits	PCC Course Equivalents	Quarter Credits
Wr 121 English Composition	4	**WR** 121 English Composition	4
Wr 227 Technical Writing	4	**WR** 227 Technical Writing	4
Comm 220 Public Speaking	4	**SP** 111 Fundamentals of Speech	4
CS 140U is not required but, students are expected to know the information in this class	-	**CS** 140U Introduction to Unix *Will transfer to PSU as a lower division CS elective*	4
CS 160 is not required but, students are expected to know the information in this class	-	**CS** 160 Exploring Computer Science *Will transfer to PSU as a lower division CS elective*	4
CS 161 is no longer required but, students are expected to know the information in this class	4	**CS** 161 Computer Science I *Will transfer to PSU as a lower division CS elective*	4
CS 162 Introduction to Computer Science II	4	**CS** 162 Computer Science II *(see note 1)*	4
CS 163 Data Structures	4	**CS** 260 Data Structures	4
CS 202 Programming Systems	4	**CS** 261 Programming Systems	4
CS 201 Computer Systems Programming	4	**CS** 201 Computer Systems II	4
CS 250 Discrete Structures	4	**CS** 250 Discrete Structures I or MTH 231 Discrete Math I *(see note 2)*	4
CS 251 Logical Structures	4	**CS** 251 Discrete Structures II or MTH 232 Discrete Math II *(see note 2)*	4
Mth 251, 252, & 253 Calculus I, II, III	12	**MTH** 251, 252, & 253 Calculus I, II, III	14
Ph 211, 212, & 213 General Physics (Calculus) w/labs 214, 215, 216 or Bi 251, 252, 253 Principles of Biology or CH 221, 222, 223 General Chemistry w/labs 227, 228, 229	15	**PHY** 211, 212, & 213 General Physics (Calculus) *(see note 3)* or BI 211, 212, 213 Principles of Biology or CH 221, 222, 223 General Chemistry	15
Approved Science Elective- 4 credits chosen from Biology, Chemistry, Physics, Geology, or Environmental Science	4	4 credits chosen from Biology, Chemistry, Physics, Geology, or Environmental Science	4
Approved Math Electives- 1 course must be at the upper-division level (300-400)	8	**MTH** 261 Applied Linear Algebra is the only 200-level course that is an approved math elective	5
Minimum of 27 credits in the arts & letters and/or social science distribution areas for the B.S. degree	27	Choose courses from arts & humanities and/or social science at PCC or take the courses at PSU	27

Note 1: If you took CS 162 between fall 2002 and fall 2004 see a PCC CS department advisor due to changes in the curriculum.
Note 2: Both MTH 231 and MTH 232 must be taken together at PCC in order to transfer.
Note 3: PHY 212 at PCC transfers as Ph 213 at PSU. PHY 213 at PCC transfers as Ph 212 at PSU.

Meet with a PCC Academic Advisor to develop an effective transfer plan that will meet your individual needs.
PCC Computer Science Department
Sylvania Campus 503-977-4393 or 503-977-4287
Rock Creek Campus 503-614-7331 or 503-614-7604

PCC endeavors to create accurate transfer guides for students; however, requirements may change without notice. Students are responsible for working with PCC advisors and their transfer institution to ensure that their academic plan will meet requirements and timelines.

Oregon State University

Portland Community College
Transfer Center
2009-2010

Transfer Guide for:
Oregon State University
Computer Science Major

Lower Division Department Requirements for the Pre-Computer Science Major:

OSU Courses	Quarter Credits	PCC Course Equivalents	Quarter Credits
WR 121 English Composition	3	**WR** 121 English Composition	4
WR 327 Technical Writing	3	**WR** 227 Technical Writing (professional core)	4
WR 214 or WR 222 Writing in Business or English Composition	3	**WR** 123 = OSU WR 222	3
COMM 111 Public Speaking or COMM 114 Argument & Critical Discourse	3	**SP** 111 Fundamentals of Speech or **SP** 112 Persuasion/Argument/Debate	4
CS 160 Computer Science Orientation	4	**CS** 160 Exploring Computer Science	4
CS 161 & 162 Intro to Computer Sci. I & II	8	**CS** 161 & 162 Computer Science I & II	8
CS 261 Data Structures	4	**CS** 260 Data Structures	4
CS 275 Introduction to Databases	4	No PCC Equivalent	-
MTH 231 & 232 Elemts Discrete Math I & II	8	**CS** 250 & 251 Discrete Structures I & II	8
MTH 251 & 252 Differential & Integral Calc.	8	**MTH** 251 & 252 Calculus I & II	9

Additional Required Courses for Specific Options:
IS - Information Systems option **ACS** - Applied Computer Science option **CS** - Computer Systems option

ECON 201 Intro to Microeconomics	4	**EC** 201 Intro to Microeconomics (IS)	4
CS 271 Comp. Architecture & Assembly Lang.	4	**CS** 201 Computer Systems (IS, ACS)	4
PH 211&221, 212&222, & 213&223 General Physics	15	**PHY** 211, 212, & 213 General Physics (CS)	15
ECE 271 Digital Logic Design	3	**ENGR** 171 Intro to Logic Design (CS) *Prerequisite ENGR 221*	4
ECE 375 Computer Organization & Computer Design	4	**ENGR** 275 Microprocessor Systems (CS) *Prerequisite ENGR 171*	4
MTH 254 Vector Calculus I	4	**MTH** 254 Vector Calculus I (CS)	5

The **Information Systems Option** includes the core classes in computer science, but also incorporates a minor in business and entrepreneurship. In addition to providing a solid grounding in both fields, this program can position the student for entry into the four-term MBA program offered by Oregon State University's College of Business. By following both these programs, the student can complete both a BS in computer science and an MBA in five years. (Note: admission into the MBA or any graduate program is predicated by acceptable grades in undergraduate coursework, an acceptable score on a standardized graduate school admissions test, and letters of reference.)

The **Applied Computer Science Option** is designed for students who wish to combine the study of computer science with an in-depth examination of a related field. This option may include a major or minor from that related field. Examples of approved programs include the multimedia minor, simulation and game programming, and software systems for sustainability and renewable energy. **Students must have an approved applied program before the end of their second term in the professional program.**

The **Computer Systems Option** is for those students who wish to take up computer science as a career, and who seek an in-depth understanding of computer science as an academic discipline. This option provides excellent preparation for those who plan to further their studies by pursing an advanced degree (masters or Ph.D.) in computer science, or for working for companies that require advanced mathematics, computing theory, or hardware design. This option is ABET/CAC accredited.

Oregon Institute of Technology

Portland Community College
Transfer Center
2009-2010

Transfer Guide for:
Oregon Institute of Technology
Software Engineering Technology

Lower Division Department Requirements for the Computer Science Major:

Oregon Institute of Technology Course Number & Title	Qtr Credits	Portland Community College Course Number & Title	Qtr Credits
CST 102 Intro to Computer Systems	3	**CS** 160 Exploring Computer Science	4
CST 116 C++ Programming I	4	**CS** 161 Computer Science I	4
CST 126 C++ Programming II	4	**CS** 162 Computer Science II	4
CST 240 UNIX	3	**CS** 140U Introduction to UNIX	4
CST 130 Computer Organization	4	No equivalent at PCC	-
CST 131 Computer Architecture	3	**CS** 201 Computer Systems	4
CST 211 Data Structures	4	**CS** 260 Data Structures	4
CST 136 Object Oriented Programming with C++	4	**CS** 261 Programming Systems	4
MATH 111 College Algebra	4	**MTH** 111C College Algebra	5
MATH 112 Trigonometry	4	**MTH** 112 Elementary Functions	5
MATH 251 Analytic Geometry and Diff. Calculus	4	**MTH** 251 Calculus I	4
MATH 252 Integral Calculus	4	**MTH** 252 Calculus II	5
MATH 254N Multivariable and Vector Calculus	4	**MTH** 254 Vector Calculus I	5
MATH 327 Discrete Mathematics	4	**CS** 250 Discrete Structures I*	4
PHY 221, 222, 223 General Physics with Calculus	12	**PHY** 211, 212, 213 General Physics (Calculus)	15
WRI 121 English Composition	3	**WR** 121 English Composition	4
WRI 122 English Composition	3	**WR** 122 English Composition	4
WRI 227 Technical Report Writing	3	**WR** 227 Technical Writing I	4
SPE 111 Fundamentals of Speech	3	**SP** 111 Public Speaking	4
PSY 201 General Psychology	3	**PSY** 201 Introduction to Psychology or **PSY** 201A Introduction to Psychology	4
SPE 321 Small Team & Group Communication	3	**SP** 215 Small Group Communication*	4
Social Science Electives	6	See General Education sheet for list of courses	8
Humanities Electives	9	See General Education sheet for list of courses	9-12
Technical Electives	9	Courses offered through PCC that can be used as technical electives, with advisor approval: **CS** 233G Game Programming **CS** 233U Advanced C Programming **CS** 251 Discrete Structures II **MTH** 253 Calculus III **	 4 4 4 4

* Lower-division course (100-200 level) satisfies the OIT course requirement but will not count toward upper-division (300-400 level) requirements.

Algebra-based General Physics PHY 201, 202, 203, may be substituted with OIT Advisor consent.

Extra credits earned at PCC and not specified toward a particular course at OIT will be shown as elective credits on OIT transfer evaluation.

**Course can be used as technical elective course with advisor approval and can also be applied toward a Minor in Mathematics.

PCC endeavors to create accurate transfer guides for students; however, requirements may change without notice. Students are responsible for working with PCC advisors and their transfer institution to ensure that their academic plan will meet requirements and timelines.

University of Oregon

Portland Community College
Transfer Center
2009-2010

Transfer Guide for:
University of Oregon
Computer Science

Recommended major requirements which can be completed at Portland Community College. This sheet should be used with the UO Direct Transfer Plan or **Associate of Arts Oregon Transfer degree**. Consult the program advisor/counselor for comprehensive curriculum planning.

Required major classes that can be taken at PCC:

U of O Courses	Credits	PCC Course Equivalents	Quarter Credits
MTH 251, 252, 253- Calculus 1, 2, 3	4,4,4	**MTH** 251, 252, 253- Calculus 1, 2, 3	4,5,5
MTH 231, 232[1]- Discrete Math 1, 2	4,4	**MTH** 231, 232[1]- Discrete Math 1, 2	4,4
CIS 210- Computer Science I	4	**CS** 161- Computer Science I	4
CIS 211- Computer Science II	4	**CS** 162- Computer Science II	4
CIS 212- Computer Science III	4	**CS** 260- Data Structures I	4
Select **one** Science sequence[2] from the following:		Select **one** Science sequence[2] from the following:	
1. BI 211, 212, 213 General Biology	4,4,4	1. **BI** 211, 212, 213 Principles of Biology	5,5,5
2. CH 221, 222, 223 General Chemistry	4,4,4	2. **CH** 221, 222, 223 General Chemistry	5,5,5
3. PHYS 201, 202, 203 General Physics	4,4,4	3. **PH** 201, 202, 203 General Physics	4,4,4
4. PHYS 251, 252, 253 Found. of Physics I	4,4,4	4. **PH** 211, 212, 213 General Physics (Calculus)	5,5,5
5. Psychology Sequence[3]		5. **PSY** 201, 202[3] Intro. to Psychology	4,4

- Students who declare Computer and Information Science as their major are considered Pre-Majors. On completion of the following courses: CS 161, CS 162, CS 260, Math 231, and Math 232, **students must formally apply for admission to the Computer Science program**. Students must achieve a GPA of 2.60 or better in these courses and the courses must be taken for a grade of C- or better. Courses can be retaken once, but all grades will be used to compute GPA.
- All required CS and Math courses listed in the note above must be taken for a letter grade and passed with a C- or better.

1 MTH 233 is not offered at PCC. Students transferring to the UO will need to take it at the first available offering.
2 Individual tracks (specialized courses of study) at the UO may have specific Science requirements. Please check the UO web page http://www.cs.uoregon.edu/education/undergrad/CIS_Major.php for track information.
3 Students fulfilling the Psychology sequence will also need to complete, at the UO, PSY 302, 303, and at least eight credits from the experimental and physiological fields (PSY 430-468).

References

- ACM 2008
  - *Computer Science Curriculum 2008: An Interim Revision of CS 2001*, © 2008 IEEE & ACM

- Becerra-Fernandez 2010
  - *Reversing the landslide in computer-related degree programs*, by Irma Becerra-Fernandez, Joyce Elam, and Susan Clemmons. © 2010 Communications of the ACM, February 2010, Vol 53, No. 2

- Coder 2009
  - *Increasing gender diversity in the IT workforce*, by LeAnne Coder, et. al. © 2009 Communications of the ACM, May 2009, Vol. 52, No. 5

- Felleisen 2009
  - *Why computer science doesn't matter*, by Matthias Felleisen, et al. © 2009 Communications of the ACM, July 2009, Vol. 52, No. 7

- Stephenson 2009
  - *It is a pivotal time for K-12 computer science*, by Chris Stephenson, © 2009 Communications of the ACM, December 2009, Vol. 52, No. 12

- Stroustrup 2010
  - *What should we teach new software developers? Why?* by Bjorne Stroustrup, © 2010 Communications of the ACM, January 2010, Vol. 53, No. 1

- Taulbee 2009
  - 2007 – 2008 Taulbee Survey, edited by Stuart Zweben, © 2008 Computing Research News, May 2009

- Zyda 2009
  - *Computer Science in the conceptual age*, by Michael Zyda, © 2009 Communications of the ACM, December 2009, Vol 52, No. 12

Note: Quantitative program data provided by PCC Institutional Effectiveness.