Clackamas Community College

Online Course/Outline Submission System

Consent Agenda Requests

Print | Edit | Delete | Back | Reject | Publish

Section #1 General Course Information

Department: Computer Science

Submitter

First Name: Jen
Last Name: Miller
Phone: 3138
Email: jen.miller

Course Prefix and Number: CS - 201

 # Credits: 4

Contact hours

Lecture (# of hours): 44
Lec/lab (# of hours):
Lab (# of hours):
Total course hours: 44

For each credit, the student will be expected to spend, on average, 3 hours per week in combination of in-class and out-of-class activity.

Course Title: Computer Systems II

Course Description:

Introduction to computer systems from a software perspective. Topics include: Basic machine organization, system programming in C and assembly language, introduction to system programming tools (gcc, makefile, gdb), data representation (bits & bytes, characters, integers, floating point numbers), implementation of control flow, procedure calls, and complex data types at the machine level, linking and loading, exceptions and interrupts, process control and signals, system calls, file I/O, timing and improving program performance, basic memory hierarchy, and dynamic memory allocation techniques.

Type of Course: Lower Division Collegiate

Reason for the new course:

This new course is required for completion of the AS in Computer Science degree. CCC Students pursuing an AS in Computer Science currently have to take this course at PCC or PSU.

Is this class challengeable?

Yes

Can this course be repeated for credit in a degree?

No

Is general education certification being sought at this time?

No

Does this course map to any general education outcome(s)?

No

Is this course part of an AAS or related certificate of completion?

Yes

Name of degree(s) and/or certificate(s): AS in Computer Science

Are there prerequisites to this course?

Yes

Pre-reqs: CS-162

Have you consulted with the appropriate chair if the pre-req is in another program?

No

Are there corequisites to this course?

No

Are there any requirements or recommendations for students taken this course?

No

Are there similar courses existing in other programs or disciplines at CCC?

No

Will this class use library resources?

No

Is there any other potential impact on another department?

No

Does this course belong on the Related Instruction list?

No

GRADING METHOD:

A-F or Pass/No Pass

Audit: Yes

When do you plan to offer this course?

✓ Fall

Is this course equivalent to another?

If yes, they must have the same description and outcomes.

No

Will this course appear in the college catalog?

Yes

Will this course appear in the schedule?

Yes

Student Learning Outcomes:

Upon successful completion of this course, students should be able to:

1. describe basic computer system organization including the operating system (processes, files, virtual memory) and the underlying hardware (CPU, registers, memory hierarchy);
2. describe the compilation system (preprocessing, assembling, compiling, and linking) and the function of object/executable files and shared libraries, as well as how basic system utilities such as debuggers and Makefiles work;
3. write C programs to illustrate basic systems programming concepts, including file I/O, system calls, memory management, exception handling and process management;
4. perform arithmetic in hexadecimal, decimal, octal, and binary notation, and convert among these notations;
5. explain how data types such as integers, characters, floating point numbers, arrays, pointers, and structures are represented;
6. describe the basic instruction set architecture for the IA32 family (or similar machine), including the arithmetic/logic instructions, registers, memory model and addressing, and control instructions;
7. explain how high-level programming constructs such as loops and stack-based function calls are implemented in underlying machine code,
8. explain how exceptions, traps, and context switches occur and how they are handled at the machine level;
9. explain the performance impact of hardware features such as pipelining, and architecture principles such as memory locality;
10. use profiling and timing facilities to identify performance bottlenecks in C programs.

This course does not include assessable General Education outcomes.

Major Topic Outline:

1. Exceptions and interrupts.
2. Processes and process control.

3. File I/O and system calls.
4. Performance measurement.
5. Performance improvement.
6. Memory hierarchy.
7. Memory allocation.

Does the content of this class relate to job skills in any of the following areas:

1. Increased energy efficiency	**No**
2. Produce renewable energy	**No**
3. Prevent environmental degradation	**No**
4. Clean up natural environment	**No**
5. Supports green services	**No**

Percent of course: 0%

Section #2 Course Transferability

Concern over students taking many courses that do not have a high transfer value has led to increasing attention to the transferability of LDC courses. The state currently requires us to certify that at least one OUS school will accept a new LDC course in transfer. Faculty should communicate with colleagues at one or more OUS schools to ascertain how the course will transfer by answering these questions.

1. Is there an equivalent lower division course at the University?
2. Will a department accept the course for its major or minor requirements?
3. Will the course be accepted as part of the University's distribution requirements?

If a course transfers as an elective only, it may still be accepted or approved as an LDC course, depending on the nature of the course, though it will likely not be eligible for Gen Ed status.

Which OUS schools will the course transfer to? (Check all that apply)

✓ **PSU (Portland State University)**

Identify comparable course(s) at OUS school(s)

CS-201

How does it transfer? (Check all that apply)

✓ **required or support for major**

:

First term to be offered:

Specify term: Fall 2015