

**Clackamas Community College**

## Online Course/Outline Submission System

**Published Requests**

CS-251 Discrete Structures II

General education certified:  Yes  No

- Writing  
 Oral Communication  
 Arts and Letters  
 Science & Computer Science  
 Mathematics  
 Social Science  
 Cultural Literacy  
 Health & Physical Education

 Approved Date (mm/dd/yyyy):  /  / **Section #1 General Course Information****Department:** Business & Computer Science: Computer Science**Submitter**

First Name: Jen

Last Name: Miller

Phone: 3138

Email: jen.miller

**Course Prefix and Number:** CS - 251**# 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:** Discrete Structures II

**Course Description:**

Continuation of the introduction to discrete structures and techniques for computing started in CS-250. The course, which is the second in the two-term sequence, aims to convey the skills in discrete mathematics that are used in the study and practice of computer science. Topics include: Logic: propositional calculus, first-order predicate calculus; Formal reasoning: natural deduction, resolution; Applications to program correctness and automatic reasoning; Introduction to algebraic structures in computing.

---

**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 Degree in Computer Science with PSU

**Are there prerequisites to this course?**

**Yes**

**Pre-reqs:** CS-250

**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?

**Yes**

Have you talked with a librarian regarding that impact?

**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?

**✓ Spring**

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?

**No**

Will this course appear in the schedule?

**No**

Student Learning Outcomes:

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

1. apply the properties of propositional calculus to determine whether a wff is a tautology, a contradiction, or a contingency by truth tables and by Quine's method, construct equivalence proofs, and transform truth functions and wffs into conjunctive or disjunctive normal form;
2. describe the basic inference rules and use them to write formal proofs in propositional calculus,
3. apply the properties of first-order predicate calculus to determine whether a wff is valid, invalid, satisfiable, or unsatisfiable, construct equivalence proofs, and transform first-order wffs into prenex conjunctive or disjunctive normal form;

4. describe the rules of inference for quantifiers and use them along with the basic inference rules to write formal proofs in first-order predicate calculus,
5. write formal proofs in first-order predicate calculus with equality,
6. construct partial correctness proofs of simple imperative programs and construct termination proofs for simple loops,
7. transform first-order wffs into clausal form and unify atoms from a set of clauses,
8. describe the resolution inference rule, use it to write formal proofs in first-order logic, and describe how resolution is used to execute a logic program;
9. transform simple English sentences into formal logic (propositional, first-order, or higher-order);
10. apply appropriate algebraic properties to simplify Boolean expressions, simplify regular expressions, write recursive definitions for simple functions in terms of operations for abstract data types, write expressions to represent relations constructed in terms of operations for relational databases, and work with congruences.

---

***This course does not include assessable General Education outcomes.***

---

**Major Topic Outline:**

1. Propositional logic: propositional calculus, normal forms, formal reasoning.
2. First-order logic: first-order predicate calculus, equivalence, quantifier inference rules.
3. Applied Logic: equality, program correctness, higher-order logic.
4. Automatic Reasoning: clausal forms, unification, resolution.
5. Algebraic Structures: Boolean algebra, abstract data types, relational algebra, congruences.

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

- |                                      |           |
|--------------------------------------|-----------|
| 1. Increased energy efficiency       | <b>No</b> |
| 2. Produce renewable energy          | <b>No</b> |
| 3. Prevent environmental degradation | <b>No</b> |
| 4. Clean up natural environment      | <b>No</b> |
| 5. Supports green services           | <b>No</b> |

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-251

How does it transfer? (Check all that apply)

required or support for major

:

First term to be offered:

**Specify term:** Spring 2016

---