CIS 201 Computer Science I

Textbook:	Recommended: "*Building Java Programs*, 3rd Edition (2014)" by Reges, Stuart & Stepp, Marty", ISBN-13: 978-0-13-336090-5. ISBN-10: 0-13-336090-3.
Instructor:	Dr. Supraja Gurajala
Office Hours:	MW 10:00am to 12:00 noon
Class Time/Place:	MWF 12:00 noon – 12:50 PM (3 contact hours) / Dunn Hall 102
Lab Time/Place:	M 1:00 – 2:50 PM (2 contact hours) / Dunn 302
Credits:	Lecture: 3 credits; Lab 1 credit (Note: concurrent enrollment in lecture and lab is required)
Prerequisites:	None

Course Description:

> **CIS 201 Computer Science I.** Introduction to computer science and systems through problem definition, decomposition, solution design, programming, and testing. Programming includes using data types, control structures, and functions; solutions will use simple data, lists, and files. The digital, binary, and general-purpose nature of a modern computer is examined by looking at the parts of the computer and how some data and instructions are encoded in it. Laboratory required. Fall and Spring.

Student Learning Outcomes:

Upon finishing this course, students should be able to:

1. Apply the appropriate levels of abstraction and top-down decomposition to break large problems into smaller, more tractable problems, solve them using single-purpose functions, sequence, selection, and iteration, recombining the smaller solutions into a solution for the larger problem.
2. Implement designed solutions into programs that use simple variables, arrays of simple variables, parameterized functions, screen input/output, and text file input/output appropriately.
3. Explain the connection between starting early, problem decomposition, and asking appropriate questions and accurate estimation of time required for a project and being able to get done.
4. Describe at least two resources used by running algorithms, including the units in which they are measured, and describe at least two levels where performance could be improved.
5. Identify the parts of a modern computer and describe what parts participate in the fetch-decode-execute and load-process operations.

Course Requirements and Grading:

1. ***Weekly Quizzes: 10 %***

   A quiz will be given starting the second lecture of the course. Quizzes take material from previous lectures, labs, meaning, if you don't finish the lab, or revise lecture materials you may not be prepared for the quiz. This will also be counted as your attendance.

2. ***Programming Assignments: 25 %***

   Several programming assignments will be given based on the concepts discussed in lectures. These programming assignments will be the essential part of the course. Programming assignments will be posted online along with the due date. Late assignments are penalized at 20% per calendar day that they are late. Your final submitted assignment should represent your individual work; it is, however, acceptable to discuss the solution approach with other students. You will be responsible for keeping track of programming assignments due dates posted online.

3. ***Exams: 40%***

   a. Midterm 1 – 13 %
   b. Midterm 2 – 13%
   c. Final Exam – 14 %

   Exams are closed book and closed notes. Any request for re-grading must be received in email and within 3 days of receiving your graded exam back. Prior notice must be given to your instructor if you can't make it to exam. No make-ups will be granted unless satisfactory documentation is produced to show an extenuating circumstance.

4. ***Labs: 25%***

   - ***Lab instructions:*** Labs will be available on Moodle each week under lab course page
   - ***Making the Best Use of Lab:*** Each lab will be based on material that was covered in lecture and the online prelab reading.
   - ***To prepare for lab:***
     - Attend lectures
     - Do the assigned homework as we discuss the associated material in class.
     - Print and read the lab instructions.
   - ***Working through the laboratory exercises:*** For a given lab, you will work through the lab exercises as described in the instructions. The laboratory exercises contain checkpoints. When you reach a checkpoint, your lab instructors will check you off.
   - ***Laboratory Grade:*** The lab component of your grade is simply the percentage of checkpoints that you complete over the semester. For example, if you complete 70 out of 80 checkpoints over the semester, your lab grade will be 70/80 = 87.5%.

Course Policies

1. **Late work**

   All due dates for the course will be strictly enforced. Prior approval will be required from the instructor for any late submission, including making up missed exams.

2. **Attendance**

   Attending all lectures and labs and completing required work is crucial to your success in this course. While attendance is not graded *per se,* in-class graded work cannot be made up without prior arrangement with the instructor. In the event of absences from weekly labs, you are required to complete the missed lab work before the beginning of the next lab session. The instructor and CS tutors will be available to help you with completing labs during posted office and tutoring hours.

3. **Absences**

   As noted above, in-class graded work cannot be made up without prior arrangement with the instructor.

   Accommodation of Religious Observances: I will make reasonable accommodation for a student's religious beliefs. Please notify me within the first week of classes about any scheduled class date that conflicts with a religious observance.

4. **Academic Integrity**

   You are expected follow the "SUNY Potsdam Academic Honor Code" (SUNY Potsdam Undergraduate Catalog, https://catalog.potsdam.edu/content.php?catoid=7&navoid=566) by doing your own work on all required work for the course unless specifically directed otherwise by the professor. **Copying is strictly forbidden, regardless of the source** (online, other students). Students caught cheating will receive a grade of 0 for that evaluation. More than one offense will result in dismissal from the course and possible disciplinary sanctions by the university. Academic Misconduct definitions, procedures, due process, and student rights are described on page in the SUNY Potsdam Undergraduate Catalog, as cited above.

5. **Grade Distribution**

   At the end of the semester, I will calculate what fraction of the possible points you have earned, and your grade will be based on this distribution:

   4.0: 95 – 100%
   3.7: 90 – 94%
   3.3: 85 – 89%
   3.0: 80 – 84%
   2.7: 77 – 79%
   2.3: 73 – 76%
   2.0: 70 – 72%
   1.7: 67 – 69%
   1.3: 63 – 66%
   1.0: 60 – 62%
   0.0: <60%

Note that final grades may be determined using a class curve of the course-grade averages.

Tentative Schedule:

	Topics	*Assignment*	*Lab*
Week 1	syllabus, class policies, parts of programs, compilation, execution, programming language, elements of a computer, using linux/terminal(lab), identifiers & keywords	Assignment 1	Lab 01
Week 2	errors, data types, escape characters, algorithm, structured algorithm, methods: header, calling, code block, & comments/javadoc	Assignment 2	Lab 02
Week 3	methods: task decomposition, reducing redundancy, methods called from other methods, control flow, using methods in drawing figures, data types, expressions, operators: arithmetic, modulo, integer division	Assignment 3	Lab 03
Week 4	variables: declare, initialize, assign, modify, print, memory use, error messages; loops: common elements in all loops, while loop, (do-loop), flow charts and pseudo code for loops, for-loop	Assignment 4	Lab 04
Week 5	loops: printing and indefinite vs definite: nested loops (simplest ones only), convert each loop type to the others, simple loop table concept: printing sequence of numbers given a specific series of indices; more pseudocode & top down design	Assignment 5	Lab 05
Week 6	Review & Midterm 1	Assignment 6	Lab 06
Week 7	scope, class constants, string objects, parameter passing: syntax, loop examples, multiple parameters, use of memory	Assignment 7	Lab 07
Week 8	return values: Math class - call Math methods & use returned values; problems for base 10 in binary systems, type casting, basic objects/classes, Strings as objects, intro to Scanners	Assignment 8	Lab 8
Week 9	More Scanners: hasNext() and next() pairings, errors, Scanner as parameter; conditional execution: simple if and if-else, if-else with return, relational operators, logical operators, logic expressions	Assignment 9	Lab 09
Week 10	Review & Midterm 2	Assignment 10	Lab 10
Week 11	java debugger, cumulative algorithms (sum & product of numbers, String concatenation), decimal binary conversions, text processing, fencepost loops	Assignment 11	Lab 11
Week 12	definite vs indefinite loops, compare loop flow-charts and pseudo-code, sentinel loops and more fenceposts, random numbers, Boolean type	Assignment 12	Lab 12

Week 13	switch statement, when to return, file processing	Assignment 13	Lab 13
Week 14	more file processing, arrays: 0-based index, two initialization types, access & modify elements, exceptions, traversal	Assignment 14	Lab 14
Finals	Final Exam, comprehensive/cumulate, time/date TBA		