

ICSI 201 - Introduction to Computer Science

Fall 2020

INSTRUCTOR OF RECORD

Dr. Vladimir Kuperman (vkuperman@albany.edu).

Office hours: I will be available for Zoom meeting every Friday 10 am – noon (use *Office Hours* link posted on Blackboard), or by appointment (send me email).

TEACHING ASSISTANTS

Name	Contact Info	Office hours (location, time)
TBD		
TBD		

TEXTBOOK

Required:

- **Starting Out with Java: From Control Structures through Data Structures, 4th edition, Gaddis & Muganda ©2019 Pearson.**

It is acceptable if you prefer 3rd edition or online version of this textbook. The advantage of having this textbook in your library is that it is required for one more Computer Science core course—ICSI 213 Data Structures.

Additional reading materials will be provided during the course. Intensive use of relevant online resources is expected in many assignments. One of the most important is Java API Specification (Java Library):

<http://docs.oracle.com/javase/8/docs/api/index.html>.

EQUIPMENT AND SOFTWARE ENVIRONMENT

1. It is programming intensive course, and you must have access to a computer and be able to install applications as needed. A personal laptop with Windows or Mac OS would be the right match.
2. The course is fully online, and you are expected to have Internet access readily available when you begin the course. In some cases, even if you have Internet access, you might need to view video and audio files on a computer that has a faster Internet connection. While other electronic devices (e.g., smartphones) may provide access to some components of the course, these are not recommended and will not work for all required activities.
3. Blackboard is an essential technology we will use to manage this course (<https://blackboard.albany.edu>). Discussion and communication tools will be widely used. UAlbany recommends Firefox as a browser that works best with Blackboard (you can download Firefox for free at

<http://www.mozilla.org/en-US/firefox/new/>). The Chrome browser works well most of the time, though. If you are unfamiliar with Blackboard, read Blackboard Help (<https://wiki.albany.edu/display/public/askit/Blackboard+Help>). If you have reviewed the help and still have questions or experiencing problems with Blackboard, please contact the help desk by phone (518) 442-3700 or by email its@albany.edu.

4. Another tool important for live online communication is Zoom. The Zoom links will be posted on Blackboard. If you have not activated your UAlbany Zoom account, please do it (<https://albany.zoom.us/>). You **must be signed in** to your UAlbany Zoom account each time we meet in Zoom.
5. The Java programming language will be used in the course. You need Java SE Development Kit 8 installed on your computers. It is free and can be downloaded from <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>.
6. In most cases for writing, compiling, and debugging programs, the integrated development environment (IDE) is recommended. We recommend Eclipse (link to download: <http://www.eclipse.org/downloads/packages/release/Neon/3>). If you cannot install Eclipse on your computer for some reason, you may use more basic IDE like Dr. Java (<http://www.drjava.org>) or even an online Java compiler, but the last option is not recommended. All mentioned IDEs are free.

COURSE OVERVIEW

This course is designed to introduce the computer science discipline, which begins with computer programming, to its majors and minors. It is also an option for some requirements in certain other majors or minors. Unlike introductory courses that are broad, shallow or conceptual introductions to various aspects of a field, this course achieves its purposes by concentrating on one central skill: computer programming. In this course you will learn to create programs in the Java language. The basics are programming the computer by writing, testing, and debugging code that you write in Java to store and process data.

You will learn to perform basic internal calculations and utilize the data storage capabilities of today's computers. We will cover basic issues of computer science, such as how to use computation for interesting applications, and computational efficiency. You will be required to employ the most basic information technology skills for programmers, learning them as necessary from course staff and classmates.

This course requires studying materials located by URLs on the public Web, writing and editing programs or other text files, compiling and running programs, web up/downloading, managing folders/directories of your files, making archives, transferring archives, files, and folders/directories between computers, keeping backups and version histories, etc.

This is a hands-on, cumulative skill practice and development kind of course – not a course of just learning facts. The critical thing you must do to succeed is to actively participate (not just listen, read and study). The course plan gives incentives for regular, frequent and sometimes challenging engagement with the subject's intellectual work, but also makes it more fun and easier to maintain the necessary attention.

No previous experience with computer programming is required – just a willingness to learn from working hard. However, the basic computer skills and general understanding of such terms as application, data file, operating system are required.

An inclination to use elementary mathematics and the detailed, precise thinking styles taught in mathematical sciences. Success in this course tends to increase with increasing levels and grades in college level mathematics study.

COURSE DESCRIPTION AND PREREQUISITES

This course first introduces with the elementary concepts of computer science such as CPU, memory, I/O devices and binary number system. It then focuses on developing basic programming skills. The topics include input and output data, data types, control structures, and functions/methods, arrays, procedural and object-oriented programming concepts, and program debugging and compilation. Prerequisites: general mathematical knowledge and computer skills are recommended.

LEARNING OBJECTIVES

- To understand, to some degree, what computer science is.
 - You will be able to recognize scientific and practical ideas that are precisely formulated, often with math and logic, which fit observations and measurements and can be the content of computer science.
- To gain an appreciation of computer science.
 - You will be prepared to study advanced computer science courses that will enable you to either (i) work with or (ii) become one of those experts who invent or construct software and maybe scientific ideas about computing.
- To learn and practice good techniques for writing, testing and debugging code.
 - You will be able to demonstrate fluency and problem-solving skills in computer programming. In other words, given a sufficiently detailed description of what a computer can do and a strategy for doing it, you will write a program that makes the computer do it by implementing the given strategy.
- To understand, be able to read and be able to write Object-Oriented programs using the Java programming language to achieve specific tasks.
 - You will be able to develop the confidence and creativity to construct novel solutions to problems. This is measured during the course by exposing students to handle novel programming quirks on exams, new assignments and giving you room to test your own inventions.

Expected Student Learning Outcomes:

1. When given requirements and specification students will be able to design and to develop basic- and middle-level computer programs using multiple tools.
2. Students will be able to understand programming fundamentals such as variables; expressions and assignments; primitive data types, class data types and data conversion.

3. Given programming problem students will be able to use library classes such as `String`, `Random`, `NumberFormat`, `DecimalFormat`, `Math`, wrapper classes, etc.
4. Given programming problem students will be able to choose the right conditional statements such as `if`, `if-else`, `if-else-if`, and `switch` statement; and the right loop statements such as `while`, `do-while`, and `for` statement.
5. Students will understand and be able to use in their programming solutions basic data structures such as array, array list, etc.
6. Students will be able to create their own Java classes with constructors, accessors, mutators, and other methods.
7. Students will understand advantages and implementation of Object-Oriented concepts such as information hiding, encapsulation, inheritance, and polymorphism.
8. Students will understand mechanisms of file input/output and exception handling.
9. Students will be able effectively communicate programming solutions in a written and oral format.

COURSE WEBSITE AND BLACKBOARD

Blackboard will be used to provide essential course materials, the most current syllabus, and assignment documents and no separate course website will be maintained. It is your responsibility to check Blackboard on a regularly basis to be aware about ongoing activities and the most recent updates.

ASSESSMENTS AND GRADING POLICY

Students are expected to read textbook and recommended reading materials. Short quizzes based on reading materials may be offered during the classes. Your participation in lecture and lab/discussion classes will be graded too. At the end of the semester you will be required to collaborate with two or three peers to work on the presentation on one of the topics studied in the course.

To master your programming skills there will be lab assignments, three homework assignments and two programming projects. Homework assignments and programming projects are similar. The difference is that homework assignments cover one-two programming concepts, when programming projects combines series of connected programming concepts.

While you may discuss a problem with another student, the work you submit must be your own. You may be asked questions about your work. If you cannot explain your code, you will get a zero grade.

IMPORTANT: Any student who submits copied work or any student that provides work for copying will earn a zero grade for that assignment and it is considered cheating. If your submission is borrowed in any part from the Internet or from other student, the violation of academic integrity will be reported to the Office of the Vice Provost and Dean for Undergraduate Education.

Weights in Final Grade:

Participation/Quizzes	6%
Labs	9%
Homework Assignment	12%
Midterm Exam	20%
Programming Projects	20%

Presentation	3%
Final Exam	30%

Grading Scale (rounded to the nearest whole number):

A	A-	B+	B	B-	C+	C	C-	D+	D	D-	E
93-100	90-92	87-89	83-86	80-82	77-79	73-76	70-72	67-69	63-66	60-62	Grade < 60

Note: students majoring in Computer Science must have grade "C" or better for this course.

GENERAL EDUCATION COMPETENCY

This course provides several opportunities for students to develop advanced writing, critical thinking, information literacy, and oral disclosure. In the mandatory homework, students are required to write a report that presents analysis to evaluate efficiency of solution to the problem addressed in the homework. Through this homework, students will acquire many general education competencies, such as Advanced Writing, Critical Thinking, Information Literacy and Oral Discourse. In the written report, students will not only demonstrate increasingly sophisticated writing according to the conventions of computer science, but also able to communicate clearly in writing, employing fundamental rules of usage, style, and mechanics in the context of computer science (Advanced Writing). The students will be required to present their work (in the mandatory homework) in the discussion class.

ATTENDANCE AND LATENESS

It is students' responsibility to keep up with the class and check on announcements made while they were absent. If a student misses any module *due to any reason*, it is the student's responsibility to study the material covered in that module. It will not be possible for the instructor or teaching assistant to conduct makeup classes.

Missing any assessment without a verifiable legitimate excuse will result in a grade of zero. Excused absences are absences due to family emergency, personal illness, bereavement, and athletic and academic obligations for which an authorized official will provide a written excuse.

No late submission and **no makeup** tests are allowed in this course. The only exception can be made for valid and verifiable extenuating circumstances, e.g. a major medical situation (see also <https://www.albany.edu/undergraduateeducation/attendance.php>). In such a case, it is the student's responsibility to contact the instructor ahead of time and arrange to take a makeup test at an alternate date/time. Makeup tests will be generally harder than regular ones.

If some schedule conflicts are possible because of your religious observance, please plan your work adequately and notify the instructor of record in a timely manner according to New York State Education Law ([Section 224-A](#)).

RESPONSIBLE COMPUTING

Students are required to read the University at Albany Policy for the Responsible Use of Information Technology (https://www.albany.edu/its/its_policies.htm). Students will be expected to apply the policies discussed in this document to all computing and electronic communications in the course.

STUDENTS WITH DISABILITIES

Reasonable accommodations will be provided for students with documented physical, sensory, systemic, medical, cognitive, learning and mental health (psychiatric) disabilities. If you believe you have a disability requiring accommodation in this class, please notify the Disability Resource Center (drc@albany.edu, (518) 442-5490). Upon verification and after the registration process is complete, the DRC will provide you with a letter that informs the course instructor that you are a student with a disability registered with the DRC and list the recommended reasonable accommodations.

ACADEMIC HONESTY AND OVERALL REGULATIONS

Every student has the responsibility to become familiar with the standards of academic integrity at the University. Student claims of ignorance, unintentional error, or personal or academic pressures cannot be excuses for violation of academic integrity. Students are responsible for familiarizing themselves with the standards and behaving accordingly, and faculty are responsible for teaching, modeling and upholding them. Anything less undermines the worth and value of our intellectual work, and the reputation and credibility of the University at Albany degree. Read the Standards of Academic Integrity and policies in the University Bulletin (https://www.albany.edu/undergraduate_bulletin/regulations.html).

CAUTION AND A STRONG WORD OF WARNING!!!! Plagiarism and other acts of academic dishonesty will be punished. Students are expected to submit original work. While you may discuss a problem with another student, the work you submit must be your own. Any student who submits copied work or any student who provides work for copying will earn a zero grade for that assignment. If there is more than one copying incident, the student will be graded an E for the course. As per college policy, cheating activity, including cheating in exams, quizzes, projects, etc., WILL be written up in a Violation of Academic Integrity Report (VAIR) reported to the college administration, which includes the Computer Science Chair, the College of Engineering and Applied Sciences Dean, and the Vice Provost of Undergraduate Studies. This will become a part of your permanent record. Multiple incidents will result in being expelled from the college.

Cheating in an exam or a quiz will result in a grade of zero. Borrowing a code from another student (or another team in the case of team projects) will be considered cheating. Likewise, any attempt to obtain partial or full source code from the Internet will also be treated as cheating.

TENTATIVE TOPIC SCHEDULE

The following schedule is preliminary and may be changed as the semester progresses.

	Week		Lecture	Text	Lab/Discussion	HW Due	Project Due
Tue, Aug 25	W1	1	Intro to Programming	Ch.1			
Thu, Aug 27		2	Intro to Programming	Ch.1			
Tue, Sep 1	W2	3	Basics of Java Language	Ch.2	Lab 1. Intro to Programming		
Thu, Sep 3		4	Basics of Java Language	Ch.2			
Tue, Sep 8	W3	5	Standard Output	Ch.3	Lab 2. Java Fundamentals		
Thu, Sep 10		6	Decision Structures	Ch.3			
Tue, Sep 15	W4	7	Loop Structures	Ch.4	Lab 3. Decision Structures	↓	
Thu, Sep 17		8	Loop Structures	Ch.4 ¹		HW1	
Tue, Sep 22	W5	9	Methods	Ch.5	Lab 4. Loops		
Thu, Sep 24		10	Methods and Objects. Files	Ch.5,4.10			
Tue, Sep 29	W6	11	Objects and Classes	Ch.6	Lab 5. Methods		
Thu, Oct 1		12	Objects and Classes	Ch.6			
Tue, Oct 6	W7		Midterm Exam		Lab 6. Files		
Thu, Oct 8		13	Arrays in Java	Ch.7			
Tue, Oct 13	W8	14	Arrays in Java	Ch.7 ²	Lab 7. Objects	↓	↓
Thu, Oct 15		15	ArrayList and More	Ch.7.13		HW2	P1
Tue, Oct 20	W9	16	More about Objects	Ch.8 ³	Lab 8. Arrays		
Thu, Oct 22		17	More Classes	Ch.9			
Tue, Oct 27	W10	18	Text Processing	Ch.9	Lab 9. Aggregation		
Thu, Oct 29		19	Object Relationships	Ch.8.7,10			
Tue, Nov 3	W11	20	Object Relationships	Ch.10	Lab 9. Aggregation		
Thu, Nov 5		21	Inheritance	Ch.10			
Tue, Nov 10	W12	22	More about Inheritance	Ch.10	Presentations	↓	
Thu, Nov 12		23	Exceptions in Java	Ch.11		HW3	
Tue, Nov 17	W13	24	Exceptions in Java	Ch.11	Presentations		↓
Thu, Nov 19		25	Review				P2
Thu, Nov 24	W14	26	Reserved		Reserved		

¹ Skip 4.10 Introduction to File Input and Output

² Skip 7.13 The ArrayList Class.

³ Skip 8.7 Aggregation.

COMMUNICATION

The teaching assistant (TA) appointed to your section is **the first point of contact** in case of any questions. If a problem could not be solved by the TA or a student for some reason needs to contact the instructor of record directly, please e-mail the instructor of record, copying the assigned TA on the message. Most emails are generally answered promptly, but please understand that some delay is possible. Please start the e-mail subject line with: "ICSI 201-Fall20," adding more details if necessary after it.

Your questions are expected to be well-thought-out and clearly formulated. The good practice is to provide a proposed answer that needs to be confirmed or discussed. In this case the instructor can see the way of the student's thinking and understand the problem better, if any. A bad example of communication is "I do not understand anything, please explain." It is a part of the learner's job—to realize what he or she does not understand.

Blackboard **forum** is a great place to discuss your questions, share your concerns and insights.

Even though the trend is to reduce formality, good etiquette and signs of respect are still valuable in online and in-person communication with teachers and classmates. Proof-reading is highly recommended too. A poorly written e-mails and posts have a good chance to remain unanswered.