Computer Programming I

CIS-120A-01

1 Section Information

1.1 Description

This is the first course in computer programming for computer science, information systems, science, or engineering majors. The course covers introductory programming using Java. Topics covered include algorithm development, control structures, subprograms and functions, parameter passing, and data types. Students write numerous **Java programs.**

1.2 Prerequisite

MAT 154A or MAT 154AA with a grade of ”C” or better or appropriate skills. (co-enrollment allowed)

1.2.1 Advisory

CAO 152A or equivalent knowledge of Windows.

2 Faculty Information

2.1 Name

Steven L. Richardson

2.2 Contact

2.2.1 Email

- steve.richardson.ltcc@gmail.com
- richardson@ltcc.edu

2.2.2 Phone

College- (530) 541-4600 ext. 333

3 syllabus

- Email Subject Line: Course Information
- 12 Week course
- November 23-30 No classes
- Assigned Discussions require 1 initial post and 1 reply.

Readings:

- Think Java: How to Think Like a Computer Scientist
- Introduction to Programming Using Java, Eighth Edition
- Java Programming
- The Java™ Tutorials

TJ2T

3.1 Student Learning Outcomes:

1. Design and write introductory-level computer programs in Java.
2. Compile java programs
3. Repair java compilation errors in written programs
4. Evaluate and criticize program design and code.
5. Solve written problem statements by writing computer programs which address the problem.

3.2 Instructional Methods:

This is an online course. Weekly work is outlined in the modules within our Canvas course. Your assigned work will consist of reading, videos to watch, sample programs to explore and play with as well as labs, quizzes on the reading and exercises in the reading material, and online discussions. You will also complete programming assignments.

Our Canvas course includes some 'discussion' (i.e. forum) components to which you may wish to post and interact with your fellow students. In particular there are discussions dedicated to each module's lab assignments. Feel free to ask questions for help with difficulties pertaining to the module laboratory assignments. Refrain from asking for help on your programming assignments as that violates your pledge about your work.

Course announcements contain important information about schedule, upcoming exams, completed grading and the like. Monitor these and read them.

Regular Contact From Your Instructor I will contact you at least every week via announcements, my comments within graded lab and programming assignments and via Canvas messaging. I will also regularly initiate discussions in which your participation is required. See the section on 'Discussions' below.

3.3 Exams:

There will be two midterm examinations and one comprehensive final exam. These are proctored.

You can find the dates for the three exams on the schedule above. Make appointments to have these exams proctored now! You will need to contact a facility near you which can proctor the exams for you. There may be a fee charged. Examples of proctoring facilities include your local community or other college or university. The facility you select for proctoring must be able and willing to accept delivery of your exam via email and must be able and willing to scan your completed exam and return it to me via email.

Arrange for this now!

3.4 Quizzes:

There are reading quizzes in each module.

3.5 Sections Excercises:

One of our online textbooks (Think Java 2 Trinkets) contains section exercises. You are assigned these and will submit your answers and/or links in module section exercise assignments or quizzes.

3.6 Lab Assignments:

You will be assigned several lab assignments in each module. You may collaborate with classmates on the laboratory assignments.

These are graded on a binary, one-point scale according to the following rubric:

- Does the submission compile and run and represent an authentic attempt at addressing the lab assignment? 2 points.
- Does the submission compile and run but and address the lab assignment only partially? 1 point.
- No submission. 0 points.

3.7 Programming Assignments:

There will be 1 or 2 programming assignments in each module. Assignments are pledged. This means that when you submit a programming assignment you implicitly pledge that the work is yours and yours alone. You may not collaborate with other students on these assignments.
These programming assignments are an opportunity for you to practice the art (yes: art) of computer programming. Practice mindfully, diligently and with single-minded purpose: to master the topic at hand.
The assignments are graded on a 5-point scale according to the following rubric:

- Does the submission conform to submission standards? 1 point.
- Does the submission compile and run? 1 point
- Does the program address the problem statement? 1 point
- Does the program correctly and effectively address the problem statement? 2 points

3.8 Trusted Status:

You begin this course with a 'trusted' status. This means that I believe your pledge, and will grade your assignments accordingly. If I find that you have violated your pledge (by submitting copied work, for example) you will lose your trusted status, and I will no longer grade any of your programming or lab assignments.

3.9 Feedback on Lab and Programming Assignments

I will promptly grade your submitted assignments, usually within 48 hours of the submission due date/time. I will provide critique in the form of comments within your CodeGrade or other submissions. I often will post sample solutions and address commonly-held misunderstandings or typical errors in the module discussion forums. or in announcements.

3.10 Discussions:

There are several discussions (aka forums) available on our course site in two categories:

3.10.1 Ungraded Discussions:

The 'Student Lounge' is precisely that - a place for you to get to know your classmates and post about most anything (safe for class please!) you like. It is not monitored by your instructor.
There are also discussions for each module (for example 'Module 2 Labs') in the class where you can ask module-specific questions, discuss issues, solutions and victories with your classmates and post with help for classmates who have posted with their own questions or difficulties with the module's material, reading, exercises or lab assignments. Your instructor will monitor these discussions and will reply as appropriate - but feel free to reply yourself!
Participation is important! Students who participate via discussions in online classes learn more, better and faster. Questions that are posed, discussed and, hopefully, addressed in the public discussion forums may benefit all students in our class, not just one individual as is the case with private messaging.
You will assess your class participation in module 11 when you complete and submit your self-evaluation, a component of your final grade.

3.10.2 Assigned Discussions:

These appear with the assignment icon alongside them. The 'Assigned Discussion: Module 1' is such an example. You must meaningfully post to these discussions and meaningfully reply to your classmates in these discussions as well. At minimum, you must reply to one of your classmate's posts in addition to making your initial post. You may view the assessment rubric which will be applied.

3.11 Evaluation Criteria:

Reading Quizzes	10 %
Assigned Discussions	5 %
Lab Assignments	10 %
Programming Assignments	20 %
2 Midterm Exams	30 %
Final Exam	25 %
Total	100 %

The letter grade assigned will be based on the following cutoffs:

90 % - 100 %	A
80 % - 90 %	B
70 % - 80 %	C
60 % - 70 %	D
<60 %	F

3.12 Policies:

3.12.1 Late Work:

Generally, late work is not accepted. But talk to me about it if you are having trouble getting a lab or programming assignment turned in on time.

3.12.2 Attendance:

Attendance is not measured, and no policy regarding attendance is in place.

3.12.3 Participation:

Class participation is important and is assessed as part of your self-assessment. Opportunities for participation include the many discussions the course provides. I will regularly initiate discussion topics, announce these, and expect you to participate.

3.12.4 Academic Honesty:

The work you turn in for programming assignments is pledged. This means that when you submit a programming assignment you implicitly pledge that the work is yours and yours alone. You may not collaborate with other students on these assignments.

3.12.5 Help:

I am here to help you learn: I am on your side! I want you to succeed in this course and beyond.
Use me as your resource. Do not hesitate to contact me at anytime for help. You may contact me via Canvas messaging or via email or via phone. Contact me if you have any difficulties or special needs. When you contact me I will reply within 24 hours, 48 hours over weekends or holidays.
I will monitor the module-specific discussions and will reply to posts there regularly.
Computer programming is hard! Don't tear your hair out in frustration, ask for help! Your classmates and your instructor are eager to provide help. But no one can do your learning for you - only you can.
We have a Disability Resource Canter available and I will accommodate any learning disability you may have to the best of my and the College's ability. If you find that you are lost or behind please do not hesitate to contact me.

4 Module 0

4.1 Programming is Hard!

Programming encompasses many aspects beyond functionality such as user experience, performance, security, and teamwork. It is not enough to strictly focus on one aspect alone while omitting the others. For projects of notable size and significance, it is not as simple as typing a few lines of code. It requires a lot of careful planning, designing, consideration, and team cooperation to be successful. In fact, more time is spent thinking than typing when programming, especially during long sessions of debugging.

In the end, programming is really about continuous, nonstop learning. Adaptability and constant learning are the keys to surviving this industry. We cannot expect to stay relevant if we do not do our part to keep learning. In such a volatile industry of exponential technological improvement, we have to keep up with its fast pace lest we end up in the dust.

I want to conclude this article by recognizing the hard work of all the developers around the world. To write this article, I had to reflect on the daily workflow of a team of developers. I had to look into the many aspects of programming and software development that usually go unnoticed. Since then, I have had a greater appreciation for all the software installed in my computer. To that end, I encourage everyone to thank a programmer today, regardless of experience. Where would we be without them?

Never take their hard work for granted.

4.1.1 Kari Savoca's take:

I want to set the record straight, because imposter syndrome is real:

- Programming is hard for everyone.
- It's okay if it doesn't come naturally to you.
- You can and will be successful if you figure out how you learn.

Learning something difficult, however, is beneficial in and of itself. The process is the prize. Struggling with code, while frustrating, is medicine for the mind.

`https://youtu.be/ob_GX50Za6c`

I remember learning how to ride a bike: I crashed and burned spectacularly! Prepare to do the same while learning how to program a computer, but know that your classmates and I will be around to pick you up and dust you off...

Take Syllabus Quiz Exam proctoring arrangements Locate the 4 textbooks Download Eclipse Write a Java program