

CS 142 Winter 2021

Martin Hock, mhock@tacomacc.edu

13833 CS 142 01 Java Programming for Scientists and Engineers I | 5 credits

Winter 2021 Classes 1/4/2021 to 3/15/2021 | Final Exams 3/16 to 3/19

Canvas Class Direct Link: <https://tacomacc.instructure.com/courses/2024263/>

Contents

Professor Information	2
Professor Information	2
Communication Policy	2
General Class Information for CS 142 Java Programming for Scientists and Engineers I	2
Course Modality	2
Schedule and Location	2
Catalog Description	2
Prerequisites	2
Textbooks & Supplemental Materials	2
Technology Required	3
Learning Outcomes	3
Course Learning Outcomes	3
TCC Student Policies	4
Class Policies and Procedures	4
Classroom Concerns/Disputes/Final Grade Appeal Process	4
Collaboration and Academic Dishonesty	4
Graded Items	5
Assessment Structure	5
Grading	6
Graded Item Submission Guidelines	7
Late Work	7
Incomplete Grade	7
Student Resources	7
Reasonable Accommodations for Religion/Conscience	7
Access and Accommodations	7
Safety	7
Syllabus Changes	8
Tentative Schedule	8

Professor Information

Professor Information

Name: Martin Hock

Email: mhock@tacomacc.edu (Please use Canvas for all class related messages)

Office: Online

Office Hours: Due to the ongoing health emergency, *via Zoom* by appointment

Communication Policy

My preferred method of communication with you is Canvas messages, which I check frequently. You can message me at any time, but you may not receive a response outside regular business hours. Business days are Monday–Friday, except for holidays. We can use Canvas messages to set up a Zoom meeting appointment for further discussion if needed.

Please set Canvas so that it sends you a message when I make a comment on your work. To do this, go to Canvas>Account>Notifications>Course Activities>Submission Comments. Click on the check mark for immediate notification to your home email or text, depending on what you prefer. You may also want to look at the other possible notifications and see what you would like to receive.

General Class Information for CS 142 Java Programming for Scientists and Engineers I

Course Modality

The course is online and asynchronous. This means that synchronous participation is not required, but you will have to put in frequent effort to keep up with the course.

Schedule and Location

We will be holding online class sessions on Zoom during scheduled class days (see schedule on the last page). The Zoom link is found on the Canvas course page. The sessions will be recorded and posted under the Panopto link on the Canvas course page so that all students can access them asynchronously.

Synchronous participation is not required and the sessions will be recorded. However, students have indicated that these sessions are very helpful, so please keep up with them.

Catalog Description

Using the Java programming language, students learn general principles of object-oriented programming, including how to design, implement, document, test, and debug computer programs. Topics include classes, objects, messages, expressions, decision structures, iteration, arrays, collections, events, and interfaces.

Prerequisites

MATH& 141 with a minimum grade of C or equivalent or assessment above.

Textbooks & Supplemental Materials

The course uses free Open Educational Resources which will be provided as needed within the Canvas course, including “Think Java CS 142 Edition” by Allen B. Downey and Chris Mayfield (with modifications) and “Introduction to Programming Using Java” by David J. Eck.

Technology Required

1. Canvas: we will be using this Learning Management System for sharing documents like this Syllabus, assignments, worksheets, etc. Canvas also provides forums where you can consult with other students about assignments. Your gradebook is there, too. You will be able to access Panopto-recorded class sessions from Canvas. We will also use it for electronic submission of assignments. I suggest that you **configure your Canvas account so that all important announcements are also delivered to your cell phone as text messages.**
2. You are going to need a modern computer for this class. Java is platform-independent – students have been able to use a Linux or Mac system as well as the Windows systems that I'll use. It should also be possible to use the lab or Information Commons computers, but please be aware that they are only open during limited hours. **Use a USB 3 drive to save.**
3. Java JDK (Java Development Kit): Please use JDK 8. I recommend Amazon Corretto 8, a free long term supported certified compatible JDK. By using JDK 8, you will be more compatible with the lab computers and your final project group partners.
4. We will be using the Eclipse IDE for Java Developers. The latest version which works with JDK 8 is 2020.06. Eclipse is an open-source IDE (Integrated Development Environment). It can be a little overwhelming at first, but over time, you will appreciate that it provides a powerful editor and debugger.
5. Email: Please use Canvas messages to communicate regarding the course.

If you need support using technology or learning how to use Canvas, please visit the Information Commons in Building 16 on the first floor at the end of the hallway.

Learning Outcomes

Learning outcomes are what you will know by the end of the course if you complete the requirements. These learning outcomes align with the Specialization Learning Outcomes for a specialization in Computer Science. However, this class aligns with many specializations and can help you work towards your degree, even if it's not in Computer Science. To learn more about specializations at TCC and their corresponding learning outcomes, visit the [College Catalog](#).

Course Learning Outcomes

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

1. Demonstrate the correct usage and syntax of the Java programming language. PLO: 1
2. Design, implement, document, test and debug a Java program. PLO: 1, 2, 3, 4
3. Apply the concepts of encapsulation and information hiding in program. PLO: 1
4. Apply basic programming constructs/concepts such as classes, objects, methods, expressions, decision structures, iteration, arrays, and collections. PLO: 1, 2
5. Write programs using a commonly accepted style, including indentation, capitalization, and commenting. PLO: 4
6. Write programs that implement user-interfaces reflecting current human factors expectations. PLO: 2, 4
7. Participate in a team to define, specify, implement, test and document a project of greater complexity than a single programmer could be expected to do within a specified time frame. PLO: 5

TCC Student Policies

TCC works hard to create a vibrant learning culture where you can succeed. Please refer to the [college-wide standards and policies](#) that support this important work. Some of these policies include Religious Accommodations, Withdraw/Retakes, and the Student Code of Conduct.

Class Policies and Procedures

Due Dates: All assignments are due by 11:59pm the day of the due date, unless otherwise listed on Canvas. See the Graded Item Submission Guidelines and Late Work section for more information.

Group work: Group work is an essential part of the team project. Please reach out to me if you have any concerns or reservations about working in pairs so we can discuss strategies to help you succeed.

Suspended operations: In the event of the campus closing due to inclement weather, a power outage, or another unexpected event, please check your Canvas messages for updates on how we will adjust our course schedule. You may be asked to watch videos in place of in-class time or complete an additional assignment to stay on schedule. In the event you are unable to access Canvas via the Portal, please use this direct link: <https://tacomacc.instructure.com/login/ldap>

Classroom Concerns/Disputes/Final Grade Appeal Process

If you have questions or concerns about this class, please talk to me about them. If we are unable to resolve your concerns, you may talk next with the Chair of the Computer Science, Astronomy, Physics, and Engineering (CAPE) department, Rebecca Sliger. Her office is in Building 15 Room 134 (e-mail rsliger@tacomacc.edu). The Chair can assist with information about additional steps, if needed. If you think that your final grade has been given in error, please see the [final grade appeal process](#).

Collaboration and Academic Dishonesty

This gets a little complicated in a programming course. Each of the three kinds of homework have slightly different policies. For the Canvas exercises, you should answer the questions yourself without searching for answers online or asking any people for a solution. You should only need to reference the Canvas class material and Oracle website. Please contact me if the feedback given by the online environment is confusing. I will try to respond within 24 hours but may not have time to respond before the deadline if you contact me the day the exercise is due!

For the team project, teams are expected to work together and share information, and may also contribute expertise to other teams. In addition, you may want to search on sites such as Stack Overflow for the answers to **focused questions about how to accomplish small steps of your solution, not the solution itself**. In other words, “How do I accomplish this small step in Java”. Any code should be adapted to your specific situation, paraphrased, and retyped, not copied and pasted verbatim. You should only adapt **one statement of code at a time** and understand the meaning of all the code you adapt.

For the weekly programming assignments, I expect you to discuss concepts with other students. However, I expect your submissions to be your own. **Never receive code (copy and paste, reading over the shoulder, email, download, and so forth) from other people who know about the specifics of your assignment or who implemented a similar project. Never show or send your code to other students.** You should not look at other students’ code at all. The only reason to look at another student’s code is if you are helping that student fix a specific problem or bug in their code that they asked you about first – focus on the specific problem at hand, don’t show them your code, and don’t copy their code. The assignments are designed so that you should never have to search online for answers, other than the

class material found on Canvas (textbooks, handouts, etc.) and Java documentation found at the Oracle website (including the Javadoc and Java tutorial). These resources explain and demonstrate concepts rather than give away code. Use of Stack Overflow may be problematic; on the one hand, it could contain the demonstration for a general concept you could apply, or it could contain code that you would be tempted to copy verbatim. I recommend staying away from this site for the assignments.

You may not make any use of any person or website that completes work for students, provides precise step-by-step guidance to a solution, or provides copies of solutions. The website Chegg has specifically been recognized by the Dean of Math, Science, and Engineering as a source of Academic Dishonesty. Chegg will provide information about student use to instructors so instructors can easily identify any student who uses it.

I use **JPlag**, a system developed at Karlsruhe University in Germany that finds similarities among multiple sets of source code files to detect software plagiarism. **JPlag** does not merely compare bytes of text, but is aware of programming language syntax and program structure and hence is robust against many kinds of attempts to disguise similarities between plagiarized files.

If you are working with other students and aren't sure whether the "help" you are getting crosses the threshold of copying, you may contact me, and I'll run a **JPlag** comparison for you. They will need to send their code in addition to you sending yours. **I use 90% code similarity as a threshold of unacceptable assignments.** You may get one warning at my discretion for borderline cases.

Finally, don't try to "fool" **JPlag**. Simple tricks like renaming variables, changing indenting and line breaks, rearranging sections of code, they are all understood and expected by the authors. Trying to fool the program is just stronger evidence of intent to deceive.

Students who are conscientious never have difficulty with this system. Please make sure you start your work early. It is often students who scramble at the last minute to get too much help who get in trouble.

I will not condone cheating or other forms of academic dishonesty. Cheating, and classroom behaviors reasonably associated with cheating, are a violation of the student disciplinary code and are punishable through the disciplinary procedures of the college. **If I find persuasive evidence that you have been cheating, I reserve the right to fail you in the entire course.** Instructors have an obligation to report incidents of cheating, or suspected cheating, in accordance with the procedures outlined in the Student Rights and Responsibilities Code.

As stated in the TCC Catalog, "Students are expected to be honest and forthright in their academic endeavors. Cheating, plagiarism, fabrication or other forms of academic dishonesty corrupt the learning process and threaten the educational environment for all students". The complete Administrative Procedure for Academic Dishonesty is available on the [TCC website](#).

Graded Items

Assessment Structure

The assessment structure for this course was created to assess your learning on the course learning outcomes. Each of these assessments is intended to help you learn about the principles of programming and progress toward our course learning outcomes. Table 1 shows an example of the points possible **(check Canvas for updates)** and weight of each assessment category.

- 1. Programming (50% of your grade):** I will post programming assignments on the class web site. There will be approximately one per week. The assignments will be posted in Canvas and will be handed in

electronically. The assignment may be updated with announcements on Canvas. Important: see below under [Academic Dishonesty](#) about the importance of originality in your programs. You should be prepared to explain your assignments if I ask.

Note on deadlines: Software is notorious for missed deadlines. I will accept late turn-ins, with 1 point per day of lateness, up to 10 days. **However, no submissions will be accepted more than 10 days late or after the last day of class.**

- 2. Canvas exercises (30% of the grade):** I will publish a few exercises on Canvas every week. Each exercise will either consist of a short programming problem or 1-5 questions based on code analysis or topics from class sessions and will be due at 11:59 PM the following Sunday night. I plan to administer 40 of these exercises and 10 of them can be dropped (due to a low score or incompleteness for any reason). Each of the 30 remaining exercises will count for 1% of your grade. Please only rely on your course materials, notes, and computer (Canvas, Eclipse, and built in Java documentation) to answer the exercises; please **do not** search the Internet for the answer or ask anyone for the answer (again, see [Academic Dishonesty](#)).
- 3. Team project (20% of your grade):** You will participate in a team to define, specify, implement, test and document a project of greater complexity than a single programmer could be expected to do within a specified time frame. Be thinking about what interests and challenges you as you proceed with the course. 50% of the project credit will be earned for your documentation that defines and specifies your project and the other 50% will be based on your tested, working implementation.

Table 1: Assessment Structure

Categories	Example Points Possible (check Canvas for updates)	Category Weight
Programming Assignments	500	50%
Exercises	30	30%
Team Project	100	20%
Total	130	100%

Grading

Canvas will update as submissions are graded. Because of the Academic Dishonesty policy, assignments may not be graded until all late work is received. If you think there has been an error, contact me as soon as possible to have it corrected. Final grades will be assigned using the following scale:

Table 2: Grading

Grade	Percent
A	93 to 100%
A-	90 to 92.999999%
B+	87 to 89.999999%
B	83 to 86.999999%
B-	80 to 82.999999%
C+	77 to 79.999999%
C	73 to 76.999999%
C-	70 to 72.999999%
D+	67 to 69.999999%
D	65 to 66.999999%
E	64.999999% and below

Graded Item Submission Guidelines

All submissions will be via Canvas. Each programming assignment will include a test program that you can run in Eclipse which indicates a score. **Please look for any red underlines in Eclipse which indicates that your submission is not valid and may result in no points awarded.** In addition, be aware of the Late Work and Academic Dishonesty policies. Each exercise will be composed of individual questions each of which has a score indicated in Canvas.

Late Work

Software is notorious for missed deadlines. I will accept late assignment submissions, with 1 point per day of lateness, up to 10 days. **However, no submissions will be accepted more than 10 days late or after the last day of class. The final team project submissions cannot be late.**

Incomplete Grade

“Incomplete” grades will only be available to students who were otherwise succeeding (C or better, 75% of the coursework completed), but were unable to complete the course due to documented extenuating circumstances. A signed “incomplete contract” will be necessary, and the course grade will revert to an “E” (or other grade as agreed in the contract) if work is not completed by the contract date.

Student Resources

During this quarter, you may encounter a need that goes beyond this class that affects your academic performance. We are here to help you be successful. You have access to great resources on campus and people ready to support you. Login to Canvas and access the [Student Resources](#) course to learn about some of these excellent resources.

Reasonable Accommodations for Religion/Conscience

Students who will be absent from course activities due to reasons of faith or conscience may seek reasonable accommodations so that grades are not impacted. Such requests must be made within the first two weeks of the quarter and should follow the procedures listed in the [Leave for Faith & Conscience policy webpage](#).

Access and Accommodations

Your access in the learning environment is important to me. If you have already established disability accommodations with the Access Services office, please communicate your approved accommodations to me at your earliest convenience so we can discuss your needs in this course. You can request that Access Services email your Letter of Accommodation to me, or you can provide me with printed copies.

If you have a disability or health condition that may benefit from accommodations to ensure access and support success in this course—and have *not yet* established services, please contact Access Services at (253) 460-4437 or access@tacomacc.edu.

Access Services offers resources and coordinates reasonable accommodations for students with disabilities and/or temporary health conditions. Services are established through an interactive process that begins with an intake appointment. Access Services is located in Building 7.

Safety

We care about the safety of our campus and community and all of us are needed in order to create a safe and secure learning environment. Please review the full list of safety resources available to you, which are listed on our [TCC Ready website](#). Additionally, please watch [this brief video](#) to familiarize yourself with our general safety practices. Please **program the Public Safety number into your cell**

phone (253-566-5111) or 5111 from any campus phone (If this number is inoperable due to an emergency, dial **253-495-4146**) and [sign up for TCC Alerts!](#) to receive emergency notifications to your cell phone.

Syllabus Changes

This syllabus is a contract between the student and the instructor. Nevertheless, sometimes (rarely) this contract may need to be changed. If it is, the change will be announced in class and posted on the class web site.

Tentative Schedule

Expect changes in the following schedule. If you are absent from class, it is your responsibility to check on announcements made while you were absent. TJ refers to the Think Java book, ItP or Introduction to Programming the other book by Eck.

Date	Description
Mon, Jan 4	Welcome
Tue, Jan 5	Setting Up Eclipse, Hello Food
Wed, Jan 6	TJ Ch 1/2, main method, binary
Thu, Jan 7	TJ Ch 2, dimensional analysis
Fri, Jan 8	TJ Ch 2/3, hoppy beverage
Mon, Jan 11	TJ Ch 3, method practice
Tue, Jan 12	TJ Ch 4, intro to logic
Wed, Jan 13	TJ Ch 4, parameter mystery/if
Thu, Jan 14	TJ Ch 4, more examples
Fri, Jan 15	TJ Ch 4, chained if (Island?)
Mon, Jan 18	MLK Day
Tue, Jan 19	TJ Ch 4, Farmer Problem
Wed, Jan 20	TJ Ch 4/5, Finishing Farmer
Thu, Jan 21	TJ Ch 5, while loops, multiples
Mon, Jan 25	TJ Ch 5, powers, primality
Tue, Jan 26	TJ Ch 5, while mystery
Wed, Jan 27	TJ Ch 5, drawing
Thu, Jan 28	TJ Ch 5, nested loop drawing
Fri, Jan 29	TJ Ch 6, arrays intro
Mon, Feb 1	TJ Ch 6, array sum, max
Tue, Feb 2	TJ Ch 6, array modification
Wed, Feb 3	TJ Ch 6, array mystery
Thu, Feb 4	Education Planning Day
Fri, Feb 5	TJ Ch 7, working with chars
Mon, Feb 8	TJ Ch 7, scanner, guess game
Tue, Feb 9	TJ Ch 7, game data, code maker
Wed, Feb 10	TJ Ch 7, input validation
Thu, Feb 11	TJ Ch 7, getGuess method

Date	Description
Fri, Feb 12	TJ Ch 7, counting matches
Mon, Feb 15	Presidents Day
Tue, Feb 16	TJ Ch 8, assertions mystery
Wed, Feb 17	TJ Ch 8, array written practice
Thu, Feb 18	TJ Ch 8, working with strings
Fri, Feb 19	TJ Ch 9, rectangles
Mon, Feb 22	TJ Ch 9, reference mystery
Tue, Feb 23	TJ Ch 10, person class
Wed, Feb 24	TJ Ch 10, person class, info hide
Thu, Feb 25	TJ Ch 10, circle class
Fri, Feb 26	TJ Ch 10, circle class
Mon, Mar 1	TJ Ch 10, circle class, extends
Tue, Mar 2	TJ Ch 10, inheritance mystery
Wed, Mar 3	ItP Ch 5, equals method
Thu, Mar 4	Intro to Programming Ch 5
Fri, Mar 5	Intro to Prog Ch 7 (ArrayLists)
Mon, Mar 8	Intro to Programming Ch 7
Tue, Mar 9	ItP Ch 11 (File I/O)
Wed, Mar 10	ItP Ch 11, file written practice
Thu, Mar 11	GUI Game Programming
Fri, Mar 12	Catch-up / Project Work
Mon, Mar 15	Catch-up / Project Work
Tue, Mar 16	
Wed, Mar 17	
Thu, Mar 18	
Fri, Mar 19	Final Project Due