

**Course Outline Summer 2024**  
**10 Week Session: May 29 – Aug 9**

**CSCI165 Computer Science II**  
**Remote, asynchronous**

Tompkins Cortland Community College  
170 North Street  
Dryden, NY 13053-0139

**Instructor Name:** Ken Whitener, *Professor of Computer Science*

**Email:** [hkw@tompkinscortland.edu](mailto:hkw@tompkinscortland.edu)

**Turn around:** Please allow **at least 24 hours** for me to respond. Plan accordingly and do not wait until the last minute to ask questions.

**Discord:** This is my preferred method of communication as it is much quicker than email and allows for nice synchronous communication between myself and you. We can voice chat, share screens, share files and generally be immediately productive. You can install it here <https://discord.com/> and then use the following invite code to join the TC3 CS Server. This is where I would like to direct all extra help inquiries. There is a group of great upper-level CS students on the server who enjoy helping out. This is a safe and fun space. ***There is a phone app and a desktop version. The desktop version provides the best experience.***

**Discord Invite:** <https://discord.gg/UWGFGvWkUM>

**Office Hours:** Reach out to schedule a virtual meeting or to chat on Discord or Teams

**Catalog Description:** This computer science course covers problem solving, algorithm and Object-Oriented design using the Java language and VS Code development environment. Applications will be from a variety of areas. This is **not a beginner's course** and is intended for those who have had at least a semester of programming. This course transfers to **most** four-year colleges. No previous knowledge of Java is required but programming experience in some language is expected. Prerequisite: CSCI160, MATH 120 or 3 years of regents high school math. 3 credits (2 lecture/2 lab hours per week).

**Prerequisites:** CSCI160, MATH 120 or three years of NYS Regents math

**Credits:** 3

**Texts & Additional Materials:**

- This course has no official textbook. All reading and research materials will be provided during the semester. Students will be expected to research and implement official Java documentation provided at:
  - <https://www.oracle.com/technetwork/java/api-141528.html>
  - <https://docs.oracle.com/javase/tutorial/getStarted/index.html>
  - <https://docs.oracle.com/javase/tutorial/java/index.html>

- You will need access to a computer with the following software installed:
  - **Java Development Kit:** JDK 13 minimum (or latest version. Looks like it's 21 as of Spring 24) <https://www.oracle.com/java/technologies/downloads/>
  - **Git:**
    - Windows users will need Git Bash. All demonstrations and configurations presented in class will be via BASH Shell. Windows users are expected to be able to translate these to their operating system.
    - Git Bash provides a "bash like" experience for the Windows user. This can be installed via **Git for Windows:** <https://gitforwindows.org/>
    - If you are a Linux or Mac user, your terminal is ready. Just need to install git.
    - All assignments will be disseminated via a private git repository. We will cover how to access these materials.
  - **Source Code Editor:** An industry strength source code editor like: VS Code, Sublime Text or Atom. VS Code will be used in class. Assistance will not be provided to students using Notepad. If you choose to use a tool other than the one demonstrated in class you are responsible for the configuration.

***You are expected to use a debugger.***

- Expect to download and install various programming libraries as the semester progress. For example, we will be writing **unit tests** using the JUnit framework.

**Student Learning Outcomes:** Upon successful completion of this course students should be able to understand and/or apply the following:

- Demonstrate algorithm design, implementation, debugging, and testing
- Use appropriate object-oriented programming design and architecture
- Use of abstraction, encapsulation, composition, inheritance, and polymorphism appropriately

#### **Course Modalities:**

This course is online and asynchronous. This means that there will not be any scheduled meetings between instructor and student. Students will be provided with class recordings from the Spring 2024 section of this course and any questions can be directed to the instructor via email, Teams or Discord.

- **Brightspace:** Brightspace will be used for the online gradebook, calendar and discussion forum. Both sections of the class have been combined into the same Brightspace shell. It will be our basic hub for all course related activities. You will find links to all the tools we will be using. You can go here to check due dates, participate in discussions with myself and your classmates and to keep tabs on your grade.

***You are expected to view the lecture recordings. Many important concepts and techniques will be demonstrated in the videos. These concepts will form the basis for exam questions, so it is vitally important that you understand the concepts.***

- **Discord:** The TC3 CS Discord Server is our help community. It is where you can communicate with your peers, with myself and with other CS students who have been in your shoes. You can ask questions here any time, although I will not be online 24/7. This has been a great way for students to interact and get help. It is also where I hold my remote office hours and do advising. There will be a dedicated channel for this course

where everyone can communicate, and I will also respond to direct messages. We can text chat, voice/video chat, share screens and even do **live coding** where we can collaborate on the same source code file. Your participation on this server is not required except for a single assignment where you will create an account (or use an existing account) and join the server and say hello.

In this condensed, summer format it will be greatly beneficial for you all to be active on the Discord class channel. I encourage folks to collaborate and bounce ideas off of each other. I will also be active in the class channel.

- **Git** will be used to disseminate all reading materials, assignments and source code. You will also be submitting all of your work via Git. Getting this setup and configured will be one of our first lab assignments. This will be an important **soft introduction** to an industry strength SVC (Source version control) application.

#### **Grade Access:**

All grades and feedback will be posted to Brightspace in a timely fashion. You will be able to determine how you did on any assignment, your score and any pointers by logging into Brightspace and clicking the **GRADES** link. This tool will also inform you of your overall grade in the course at any time throughout the semester. If at any time you are wondering how you are doing in the course, check your grades on Brightspace as a first course of action. You may also discuss any grade or feedback with me personally during office hours or asynchronously via email or Discord.

**Please allow 7 days turnaround for graded assignments. In most cases, it will be sooner than this but as the projects grow in complexity so does the grade time.**

#### **Grading:**

A	93-100	B -	80-82	D +	67-69
A -	90-92	C +	77-79	D	63-66
B +	87-89	C	73-76	D -	60-62
B	83-86	C -	70-72	F	0 - 59

#### **Method of Evaluation:**

**Note:** The quantity of lab assignments and homework assignments may vary due to the pace of the course

<b>Gradable Item</b>	<b>Quantity</b>	<b>Points Each</b>	<b>Total Points</b>
Midterm and Final Project	2	100	200
Midterm and Final Exam	2	100	200
Lab Assignment	11	30	330
		<b>Total Points</b>	<b>730</b>

## Explanation of assessment items:

- **Lab Assignments (30 Points):** These assignments may contain research requirements outside of the regular class readings. You **are expected** to be able to problem solve and research on your own. The reading materials will not hold your hand through the labs. You will be expected to demonstrate creative thinking and problem solving in the application of the topics. ***Students are expected to leverage debugging tools to assist them in problem solving.*** These should take anywhere from 4 – 8 hours a week. Do not procrastinate. It may take you longer.

**All questions for the instructor or teaching assistants will expect to be prefaced with**

- Specific, problem-based issues. It is difficult to provide assistance to broad generalities. ***We need to see code and talk about specifics.***
- Proof of previous problem-solving efforts. These could take the form of any of the following
  - Commented code attachments accompanied by detailed explanations of what you are attempting to accomplish.
  - Screen shots of error messages accompanied by detailed explanations of what you are attempting to accomplish.
  - Walk-throughs of electronic debugger sessions. If you are an online student, this could take the form of a screen capture video, a live screen sharing on Discord or a VS Code live coding session.
  - Links to articles, forums and technical documentation that you are trying to make sense of.
- **Lab Assignment Grades** Your lab assignments will be graded out of 30 points. To determine the spread for Midterm and Final projects, simply scale the numbers by ~3. In assessing your work, I will look at any/all of the following as appropriate:
  - **Program Logic and Structure (8 points)**
    - program demonstrates proper object-oriented programming techniques
    - program logic is efficient and uses loops when possible
    - creative use of data structures, no large blocks of if, else if
  - **Correctness/Testing (15 points)**
    - program logic solves the stated problem
    - program results are displayed to the correct precision and output formatting
    - when required, unit tests are created or adhered to.
    - tests consider all cases and boundary conditions
    - all tests pass
  - **Style (5 points)**
    - variable names are appropriate and meaningful and follow established structural conventions and are consistent
    - nested structures are properly indented
    - extreme levels of nesting are avoided
    - white space is used to improve readability
  - **Documentation (2 points)**
    - authorship is identified in code
    - program purpose is clearly stated in code
    - documentation included throughout the code as needed
    - documentation follows established standards

- documentation uses proper spelling, grammar, punctuation, etc.
- **Larger Projects:** You will have larger midterm and final project to assess your progress in the course. These projects will have a longer time span to complete. Thorough documentation will be expected
- **Exams:** You will have two exams at midterm and end of the semester. These exams will be multiple choice questions that involve code reading and concepts. They will be closed book, note, code and Internet.

**Submissions:** All submissions and course material will be fetched from a remote Git repository. There is a document explaining the software and the process and should be the very first document you read. We will be using Brightspace only for discussions/announcements and for the grade book. All files that I hand out to you will need to be fetched from the repository using the terminal. The only exception to this will be the class recordings as those files are quite large. These will be posted in Brightspace. Be sure you can find them.

#### **Attendance:**

- To maintain good grades regular attendance in class is necessary. In an online class attendance is judged a little differently. I consider a student as “attending” if they are regularly submitting assignments.

#### **Make-Up Policy / Late Work:**

- Late labs or programming assignments **will be graded with a 10% reduction in grade per day late**. Submitting an assignment past the due date/time is an automatic 10% reduction whether it is 5 minutes or 5 hours late. Plan accordingly.
- **All assignments are expected to be submitted BEFORE the specified due date.** This policy is strictly enforced.

#### **Brightspace Assistance**

Questions about your course requirements and assignments should be directed to your instructor. If you are having technical issues, like an error message when uploading a file, you can contact either the Open SUNY Helpdesk or the TC3 Tech Support Desk. You can contact the SUNY Online Helpdesk by email at [opensunyhelp@suny.edu](mailto:opensunyhelp@suny.edu) or by phone 1-844-OPENSUNY. The TC3 Helpdesk information can be found here: [https://tompkinscortland.on.spiceworks.com/portal\\_registrations](https://tompkinscortland.on.spiceworks.com/portal_registrations) or [help@tompkinscortland.on.spiceworks.com](mailto:help@tompkinscortland.on.spiceworks.com) or 607-844-8211 x 4550 or by stopping by Room 208 in the Main Building.

**Netiquette:** Classroom Civility applies in virtual class spaces as well. Typical standards of behavior that apply to on-campus behavior also apply online. These include dressing appropriately, not smoking/vaping, not consuming alcohol, and not engaging in illicit or illegal activity. The general rules of being engaged in class and being respectful apply to virtual spaces.

#### **E-mail use**

TC3 recognizes the value and efficiency of communication between faculty/staff and students through electronic mail. At the same time, email raises some issues concerning security and the identity of each individual in an email exchange. The College encourages all official student email correspondence be sent only to a student's TC3 email address and that faculty and staff consider email from students official only if it originates from a TC3 student account. This allows the College to maintain a high degree of confidence in the identity of all individuals corresponding and the security of the transmitted information.

### **Academic Integrity:**

Plagiarism and academic dishonesty are unfortunately very common in computer-related classes. It usually occurs as an act of desperation when the material is difficult, or time is tight but is nonetheless wrong and will not be tolerated in any form. My policy on academic honesty is as follows:

Any individual assignments that you submit to me in this class are a reflection of what you have learned as well as your own individual style and creativity. All individual assignments that you submit to me are expected to be your own, original work. You may not copy any work from your classmates, friends, instructors, texts, reference books, or online resources and submit it as your own. It is important that you also protect your work, and do not share any of it with your classmates. NEVER give anyone a copy of your assignment "just to look at" - it will likely be used improperly! Do not share storage devices with another student.

You are certainly encouraged to ask questions about the assignments in this course. You can ask advice on how to begin, share opinions on what process to follow to solve a specific problem, and even ask specific questions regarding troublesome aspects of the work you are completing. DO NOT, however, cut and paste large segments of an assignment for the class to see and help you with. If you need such help you may ask me. Again, the main point is to protect the work that you've created, and to allow everyone in the course an equal chance to learn the material by creating their own assignments.

Any assignments that I feel are not your original work may be checked against your classmates and other submissions from previous courses. If I determine that your submitted work has been plagiarized in any form, I will contact you immediately. On the first offense, you will receive a 0 for the assigned project, and the incident will be reported to the Dean of Students. If a second offense should happen, you will receive an F in the course, and the incident will be reported to the Dean of Students, who may choose to take further college action. Please note that all individuals involved will be sanctioned, those who give solutions as well as those who receive solutions to assignments. Remember, it is your responsibility to protect your work!

**Policy on the use of AI tools:** The release of Dall-E-Mini in July 2022 and ChatGPT in November 2022 are among many freely available tools using artificial intelligence in incredible ways. Some are calling it the **fourth technological revolution**. There is a good/great possibility that using tools like these are going to become an important skill for careers in the not distant future, and as such it is important that you begin developing these skills while also understanding the ethical implications and practicing ethical use.

<https://www.theguardian.com/commentisfree/2023/jan/07/chatgpt-bot-excel-ai-chatbot-tech>

In the meantime though, it's going to take time for society to determine when using these tools is/isn't acceptable. There are three reasons why:

1. Work created by AI tools **may** not be considered original work and instead, considered automated plagiarism. The output is generated from previously created texts/code bases that served as training sources for the models, and whose work **was not cited**. In recent months there have been lawsuits over these violations of intellectual property.

<https://techcrunch.com/2023/01/27/the-current-legal-cases-against-generative-ai-are-just-the-beginning/>

2. AI models have built-in biases (ie, the limited training data reflects society's biases; they reproduce, rather than challenge, errors in the sources). This is less of a problem for coding work but still an important concern.

<https://www.nist.gov/news-events/news/2022/03/theres-more-ai-bias-biased-data-nist-report-highlights>

3. AI tools have limitations (ie, they lack critical thinking to evaluate and reflect on criteria; they lack abductive reasoning to make judgments with incomplete information at hand) and suffer from hallucinations.

<https://www.mckinsey.com/~media/mckinsey/email/genz/2023/04/2023-04-11b.html>

Given these (important) ethical caveats, some scholars in computational sciences debate if the hype over AI-based tools-- especially as "automated plagiarism" tools-- should be heeded at all

<https://irisvanrooijcogsci.com/2023/01/14/stop-feeding-the-hype-and-start-resisting/>

For the time being, I'm modifying my academic integrity policy to include rules for responsible use of AI-based tools in class.

You may use AI programs e.g. ChatGPT as a learning assistant, to brainstorm, summarize difficult reading passages, dig deeper into terminology/concepts, and decipher error messages. However, you should note that the material generated by these programs may be incorrect, inaccurate, incomplete, or otherwise problematic. **Also understand that over reliance will stifle your own independent thinking and creativity, which is what educational institutions and employers value most.**

Academic integrity is a core principle at Tompkins Cortland Community College and it is vital that all students uphold this principle; whether using AI-based tools or otherwise. For this class, a responsible use of AI-based tools in completing coursework must be done in accordance with the following:

1. You must clearly identify the use of AI-based tools in your work. Any work that utilizes an AI tool must be clearly marked as such, including the specific tool(s) used. For example, if you use ChatGPT-3, you must cite:
  1. "ChatGPT-3. (MM/DD/YYYY of query).
  2. "Text of your query."
  3. Generated using OpenAI. <https://chat.openai.com/>"
2. You must be transparent in how you used the AI-based tool, including what work is your original contribution. An AI detector such as GPTZero (<https://gptzero.me/>) may be used to detect AI-driven work. If the source of your work is unclear, **I may require you to meet with me to explain the ideas in detail and provide inside into your thought process.**
3. You must ensure your use of AI-based tools does not violate any copyright or intellectual property laws.
4. You must not in any way use AI-based tools to cheat on assessments such as exams.
5. You must not use AI-based tools to plagiarize without citation.
6. Material generated using other tools including search engines, forums etc should follow a similar citation convention with a minimal inclusion of the url

Violations of this policy will be dealt with in accordance with our academic integrity policy, which can be found here.

[https://tc3.libguides.com/plagiarism\\_students](https://tc3.libguides.com/plagiarism_students)

**Please feel free to contact me regarding this or any other course policy if needed.**

College policies and procedures regarding academic integrity may be found in the college catalog: <http://www.tc3.edu/admissio/catalog.asp>

### **Stress Management & Emotional Wellness:**

Stress is a common part of the college experience, and it can be intensified by unexpected life changes outside the classroom. As a student, you may experience a range of issues that can cause barriers to learning, such as significant stress, mood changes, excessive worry, problems with eating and/or sleeping, strained relationships, increased anxiety, alcohol and other drug use, feeling down, difficulty concentrating, and/or lack of motivation. These issues can deteriorate your emotional and physical health, leading to diminished academic performance.

Tompkins Cortland Community College provides mental health services to support the academic success of students. The Office of Mental Health Services (OMHS) Counseling Center offers free, confidential services to help you manage personal challenges that may threaten your well-being and academic success. To speak with a clinical counselor, contact the Counseling Center at 607.844.6577 or stop by Room 230.

### **Accommodations:**

This material is available in alternative formats upon request. It is the College's policy to provide, on an individual basis, reasonable accommodations to students with disabilities, which may affect their ability to fully participate in program or course activities or to meet course requirements. Students with disabilities are encouraged to contact Carolyn Boone, Coordinator of Access and Equity Services, at (607) 844-8222 X4283 or boonec@tc3.edu to discuss their particular needs for accommodations.

### **Course Schedule/Outline (Units of Instruction):**

This summer course is a 10 week offering of 15 weeks of material. With that in mind I have decided to organize the class in the following way

All assignments for the first 5 weeks will be due Wednesday July 3. This will include

- The first 6 labs
- The midterm project
- The midterm exam

All assignments for the second 5 weeks will be due Friday August 9. This will include

- The last 5 labs
- The final project
- The final exam

My hope is that this will allow folks the freedom to be productive when they can. These assignments can be time consuming so **do not procrastinate!! The late work policy will be strictly enforced for all assignments.** You will need to be able to make it through multiple labs in a week.

If you let me know as soon as you have submitted an assignment I will promptly get you some feedback and a grade.

The following is a list of the concepts that will be covered.



Class Topic
<p>Course Overview and policies Tools Introduction to Java, program structure and execution environment The Java Virtual Machine: compiler and interpreter Introduction to Git and work submission</p>
<p>Expressions, variables and assignment statements String Class and program style Introduction to the Java API</p> <p>Console Input and Output the Scanner class</p>
<p>Flow control with branching and repetition Objects and References The String pool and string interning the equals method</p>
<p>Introduction to Object oriented programming and Design Defining Classes Instantiation and Encapsulation: access modifiers Methods</p>
<p>Classes Continued Constructors and method overloading toString and equals</p> <p>Unit Testing with JUnit</p>
<p>Programming with Arrays and ArrayLists Static vs dynamic memory allocation</p>
<p>Two Dimensional Arrays Cellular automata and matrix traversal</p> <p>Begin midterm project: Simulating the spread of forest fire</p>
<p><b>Object Oriented Design Patterns: Composition</b></p> <p>Classes Continued Feature ownership: the static keyword privacy leaks</p>
<p><b>Object Oriented Design Patterns: Inheritance</b></p> <p>Derived classes and method overriding Automatic type promotion and type casting</p>

Composition vs Inheritance
Polymorphism Dynamic Dispatch via late method binding Abstract classes and interfaces
Exception Handling Try/Catch Blocks and Exception classes  The catch or throw dilemma Designing custom Exceptions