Program cover Document – MAT 203: Introduction to Mathematical Computing

I. Basic Course Information

MAT 203: Introduction to Mathematical Computing is a 1 unit course. It will have two 80-minute meeting periods each week. It is a freshman-level course to introduce elementary programming skills and algorithmic thinking to Mathematics majors; these concepts will be applied to problems that are mathematical in nature, such as solving nonlinear equations, visualizing functions, simulating random events, etc. MAT 120 is a prerequisite for the course.

For Undergraduate Bulletin: This course introduces mathematics and statistics students to elementary concepts of both programming and computational thinking with an emphasis on mathematical problem solving. No prior programming experience is expected of students. Today's mathematicians rely on computers to solve complex problems, verify solutions, and analyze and visualize data. In this course, students will learn the foundational programming skills that are essential for performing these tasks effectively. After taking this course, students will incorporate these skills in their future mathematics and computer science courses, internships, and research experiences.

II. Learning goals

Students will gain proficiency in elementary programming and computational thinking. As the course is designed for Mathematics majors, students will apply these skills to mathematically-focused problems. The primary learning goals of this course are that students will be able to:

1. design programs to study mathematical problem statements in a well-organized fashion;
2. apply good programming practices including writing pseudocode, variable/function naming practices, and code documentation;
3. debug programs that they have created and verify that the programs they write are working properly;
4. identify the basic programming techniques used by scientists (e.g., variables, functions, conditional statements, loops) and know in which situations they can be applied; and
5. adapt their knowledge from one programming language to another.

The main topics covered include the basic elements of programming, variables, logical and conditional statements, for-loops, while-loops, basic data structures, visualization, iterative algorithms, functions, and randomness. In each module, there will be an emphasis on best practices for pseudo-code conceptualization and code debugging. These topics will be applied to a variety of mathematical problems that includes, but is not limited to, computing sequences, calculating sums and products, solving nonlinear equations (including Diophantine equations), visualizing complex functions, simulating random events (e.g., Monte Carlo applications), modular arithmetic, identifying prime numbers, and optimizing functions. Upon completion of the course, we expect students to understand the basic tenets of computer programming, write their own computer codes, comprehend existing code, and to utilize computer programming to solve mathematical problems. These performance goals are in agreement with the department's program goals.

III. Student Assessment

Students will receive regular feedback on their understanding of core programming skills and algorithmic thinking through written assignments, quizzes, projects, in-class work, and/or examinations.

IV. Learning Activities

Learning activities will consist of a combination of lectures, demonstrations, group work, student presentations, and computer assignments. The specific choice will depend upon the individual instructor. Outside of class, students are expected to do a significant amount of individual and group homeworks and projects to achieve the learning goals. These learning activities are typical of the learning activities in the department's programs.

MAT 270:
Introduction to Mathematical Computing

Syllabus
Fall 2025
1.0 Units

Instructor: Matthew Mizuhara
Office: Science Complex P-211
Email: mizuharm@tcnj.edu

Email Policy: Outside of class and office hours, e-mail is typically the fastest and best way to reach me. I try to answer e-mails within 48 hours, except on weekends. If I don't respond within 48 hours then feel free to resend the e-mail, or bring it to my attention in class.
I also expect you to check your e-mails regularly and respond in a reasonable time frame.

Course Attribute: Quantitative Reasoning

Course Materials

Textbook: There is no required textbook for this course. However, if you'd like extra practice or additional reading, I can recommend the following resources:

"MATLAB – A Practical Introduction to Programming and Problem Solving," by Stormy Attaway

"Introduction to MATLAB for Engineering Students," by David Houcque

"A MATLAB Exercise Book," by Ludmila I. Kuncheva and Cameron C. Gray

"Think Python: How to Think Like a Computer Scientist," by Allen B. Downey

Software: We will primarily use MATLAB this semester. It is freely available to all students at TCNJ. Please check the course Canvas page for instructions on downloading/accessing MATLAB from your personal laptop or from any campus computer.

Class Information

Class Times: Monday/Thursday :: 11:00 – 12:20pm

Location: Physics & Math P-218

Office hours:

Wednesday (shared) 9:00am – 11:00am
Thursday 12:30pm – 2:00pm

In-person office hours will take place in Physics & Math P-211. Office hours are a great time to go over questions from class, upcoming and past assignments/tests, etc. All office hours are "walk-in" meaning you can join at any point and leave whenever you would like. Oftentimes students join office hours and work in groups using the instructor as a resource whenever questions arise.

Special Days in Course Schedule:

Monday, September 1 – No class
Tuesday, September 2 – Class meets at regular time (college on Monday schedule)
Monday, October 6 – No class
Thursday, November 27 – No class

Course Description

The goal of the course is to introduce mathematics and statistics students to elementary programming and computers. The class will serve to expose incoming majors to the elements of basic programming structures, with no prior programming experience expected of students. After taking this course students will be able to incorporate these skills in the mathematics courses they will subsequently take as part of the major, and gain essential skills that would be beneficial in future internships and REUs (Research Experiences for Undergraduates). Programming assignments focus on problems that are mathematical in nature, giving students an opportunity to use simulations to understand and verify familiar mathematical results.

Learning Goals:

1. Students will be able to design programs to study mathematical problem statements in a well-organized fashion.
2. Students will be able to apply good programming practices including writing pseudocode, variable/function naming practices, and code commenting.
3. Students will be able to debug programs that they have created and verify that the programs they write are working properly.
4. Students will be able to identify the basic programming techniques used by scientists (e.g., variables, functions, conditional statements, loops) and know in which situations they can be applied.
5. Students will be able to adapt their knowledge from one programming language to another.

Course Expectations:

1. ***Math is learned by practicing, making mistakes, and actively engaging with the material.*** No one is an expert on day one! You should attend and participate in all class sessions. You should be brave enough to make mistakes and not value yourself or others

on right/wrong answers. If you must miss a class, then notify me as soon as possible (in advance ideally) and obtain any missed notes from a classmate.

2. ***Distractions reduce our ability to learn.*** You are welcome to use tablets to take notes if you prefer. We will also sometimes use anonymous polling which requires the use of a cell phone/laptop/tablet. However, using electronics for non-learning purposes will only distract you and hinder your ability to learn. If inappropriate use of electronics becomes an issue, then it will be reflected in your course grade.

3. ***We learn better by working in groups.*** Learning is collaborative and we learn best when working in groups of diverse backgrounds and experiences. Class time will include large and small group discussions wherein you can work together to explore new material. You are additionally encouraged to form groups outside of class for further studying!

4. ***Our classroom will be thoughtful of diversity, equity, and inclusion.*** All members of the classroom will contribute to a respectful, welcoming, and inclusive environment for all other members of the course. We all will strive to acknowledge and reduce our biases in the classroom in alignment with the TCNJ Diversity Statement (https://diversity.tcnj.edu/campus-diversity-statement/).

5. ***The instructor will be a resource for promoting equity and inclusion.*** If anyone's comfort or ability to thrive in the course is affected by any classroom or external situations, they should feel safe contacting me or the department chair to address the issue.

6. *The instructor will be organized and prompt in making course announcements and be as clear as possible about expectations.* I will post materials and course information on the Canvas page. You should check this frequently (or have it forwarded to your e-mail). I will do my best to make sure the learning objectives are clear for each topic.

Fourth Hour: In this course, the deeper learning objective of the 4th hour will be met by out-of-class coding projects. Students will work individually and in small groups to write more complex programs to synthesize and engage more deeply with course materials.

Use of AI: Leaning happens through *struggle* and *persistence.* It is normal and expected to *feel frustrated or confused along the way.* **ChatGPT and other similar LLMs can be a useful resource, but when used incorrectly will hamper your learning!** If you do not dedicate enough time to explore and make mistakes on class material and homework, then exams and quizzes will seem much more difficult later. Here are some **appropriate uses** of ChatGPT (or other generative AI):

1. Asking ChatGPT to explain a concept from class in different ways
2. Asking ChatGPT to provide extra practice problems on a certain topic
3. Asking ChatGPT to walk through new examples related to class material
4. Asking ChatGPT to walk you through solutions of previously graded assignments

Here are some **inappropriate uses** of ChatGPT (or other generative AI):

1. Asking ChatGPT to solve homework/projects
2. Asking ChatGPT to correct your homework/projects
3. Asking ChatGPT for help during group work in class

Grading/Assessments

Grading distribution:

Homework = 10%

Quizzes = 15%

Projects = 15%

Exams = 40%

Final = 20%

Grading scale:

	A : 93 - 100%	A- : 90 - 93%
B+ : 87 - 89%	B : 83 - 86%	B- : 80 - 82%
C+ : 77 - 79%	C : 73 - 76%	C- : 70 - 72%
D+ : 67 - 69%	D : 60 - 66%	F : < 60%

Homework: Homework will be assigned approximately once a week. You are encouraged to collaborate with tutors, classmates, and me to discuss homework. However, the actual completion of the homework is to be done alone. Homework extensions will be granted if appropriate notice is provided by the student. Homework may require submission of completed code, scratch work, pseudocode, or other reflections.

Quizzes: There will be quizzes every non-exam week at the beginning of class and will take approximately 20 minutes. Anyone arriving late to class or leaving class after the quiz will receive a zero. Quizzes will always be announced. *The purpose of quizzes is to check whether you are progressing appropriately between exams.*

Projects: Every few weeks we will have a larger coding project which will review and build upon the ideas learned in class. We will spend some time working on these projects in class, but they will be completed outside of class.

Exams: There will be three in-class exams. The second and third exams will not be cumulative.

Tentative exam dates: Sept 22, Oct 27, Dec 1

Final: There will be a cumulative final exam at the end of the semester. The final exam period is December 9 – December 17. The College Final Examination Policy is available here. You must contact me two weeks prior to our final exam in case of conflict.

Other Policies

The TCNJ community is dedicated to the success, safety and well-being of each student. TCNJ strictly follows key policies that govern all TCNJ community members rights and responsibilities in and out of the classroom. In addition, TCNJ has established several student support offices that can provide the support and resources to help students achieve their personal and professional goals and to promote health and well-being. You can find more information about these policies and resources at the TCNJ Student Support Resources and Classroom Policies webpage.

Students who anticipate and/or experience barriers in this course are encouraged to contact the instructor as early in the semester as possible. The Accessibility Resource Center (ARC) is available to facilitate the removal of barriers and to ensure reasonable accommodations.

https://academicaffairs.tcnj.edu/tcnj-syllabus-resources/

Tentative Calendar

Overview of course content:

1) MATLAB introduction
2) Variables
3) Functions
4) Debugging
5) Logic and conditionals
6) Vectors/Matrices
7) Plotting
8) For loops
9) While loops
10) Recursion
11) Sorting algorithms
12) Randomness
13) Comparing code to Python

Tentative Timeline (subject to change):

Week 1 : 1, 2	Week 2 : 3	Week 3 : 3,4
Week 4 : Project, Exam 1	Week 5 : 5	Week 6 : 6
Week 7 : 6, Project	Week 8 : 8, Exam 2	Week 9 : 9
Week 10 : 9, Project	Week 11 : 10, 11	Week 12 : 12
Week 13 : Project, Exam 3	Week 14 : 13	Week 15 : Project, Review

The instructor reserves the right to change the syllabus policies at any point of the semester