

**EGR111 Introduction to Computer Science (C Language)**Spring 2024

Class Schedule

Section	Days	Times	Room Location
6090	Monday, Wednesday	1:00 PM – 3:00 PM	Zoom Virtual – See Moodle for link

Instructor Information

Instructor: Keith E. Kelly
Office: (Virtual – use Zoom link provided)
Phone: 231-995-1312
E-mail: kkelly@nmc.edu

Office Hours:

Virtual Zoom office hours to be determined – see course discussions.

Course Description:

An introductory course in computer science with emphasis on C/C++ programming. Topics include structured programming, control structures, functions, arrays, pointers, memory, and compilation concepts, searching and sorting algorithms, file I/O, and top-down analysis of problems. Basic concepts of object-oriented programming will also be introduced. Group 2 course. (3 Credit Hours /4 Contact Hours)

Prerequisite Courses / Placement:

MTH 111 - may be taken concurrently.

Teaching Methods

We will use lectures, homework, activities, and projects to understand course content. Coding activities will be completed using standard software developer tools running on a Linux-based Raspberry Pi single board computer. This device is required and must be purchased from the NMC bookstore. Instructional content is provided using open-source resources. A course textbook is not required.

This is an online synchronous course. This means that the course is delivered in a video conferencing format using the Zoom video conferencing software. You are required to attend virtual class sessions and attendance will be tracked. You will need access to a computer, a quality Internet connection, and a microphone. A video camera is also required. I expect students to have video cameras on during sessions.

Required Course Material:

Textbook: A textbook is not required for this course. Links to open-source resources are provided.

RPi Kit: A Raspberry Pi single board computer kit is required and available for purchase in the NMC bookstore. Please contact me if you are remote. We'll arrange to ship the kit to you.

Computer

Internet connection

Video camera

Microphone

Wi-Fi connection

Ethernet connection (this may require a USB adapter cable for your laptop)

Course Objectives / Learning Outcomes:

- Write programs using data and control structures.
- Create C Language algorithms and solutions using standard industry tools.
- Create structured solutions using functions and scoping.
- Design solutions to experience and value statically typed, compiled computer languages.
- Access and process data using arrays, pointers, file I/O, and data structures.

Course Outcomes:

Area	Learning Outcome	Assessment Tool
Knowledge	Students will effectively use the basic syntax and semantics of the C/C++ programming language. They will use an integrated development environment to edit, compile, test, and commit coding projects.	Quizzes, Exams, End of Sprint Projects
Application	Students will design and implement C/C++ programs using procedural computer programming language supporting structured programming within a static-typed system.	End of Sprint Projects
Integration	Students will identify a problem, visualize a solution, identify requirements, determine tests, code, validate, and publish.	End of Sprint Projects
Human Dimension	Self: Students will strive to improve code quality based on feedback. Others: Students will complete pair programming and code review activities.	Classroom activities, Peer Coding
Caring – Civic Learning	Students will recognize the impacts of code and data on both business and individuals including social and ethical aspects.	Coding/Solutions - Reflections
Learning How to Learn	Students will research and use programming patterns and practices not covered in class.	Project Extensions

General Education Outcomes:

Critical Thinking:

Students will skillfully conceptualize, apply, analyze, synthesize, and evaluate information gathered from observation, experience, reflection, reasoning, or communication.

Grade Determination:

Final grades will be determined as follows: Total of all deliverables including tests, quizzes, worksheets, homework, projects, and exams divided by the total possible points x 100%

Grading Scale:

4.0 = 93% or above

3.5 = 85 – 92%

3.0 = 80 – 84%

2.5 = 75 – 79%

2.0 = 70 – 74%

1.5 = 65 – 69%

1.0 = 60 – 64%

0.0 = below 60%

Proposed Assignments / Grading Criteria:

<i>Assignment</i>	<i>Points</i>	<i>Percentage of Final Grade</i>
Weekly assignments	400	40%
Projects (4)	400	40%
Exams (2)	200	20%
TOTAL	1000	100%

Attendance/Participation

You are expected to attend each class session and to actively participate in class by asking questions, working on in-class exercises, giving presentations as individuals or as part of team projects, and sharing experiences and opinions related to the topics discussed. Students who do not participate in class or miss more than four in-class hours without a pre-approved excuse will have their final grades reduced by one grade (i.e., 4.0 to 3.5). Be sure to contact me BEFORE you miss a class, if possible. Extended or initial absence can result in the instructor dropping you from the course.

Let me know about last minute emergencies via email or phone as soon as you can.

Late Work

Work must be submitted by the stated deadline. There is an opportunity to make-up missed points at the end sprints. The instructor will discuss the concept of technical debt and the process for making up points. This policy applies only to production and project work. Exam points cannot be made-up. See the course web site for descriptions of homework assignments. If you have a special circumstance, let me know in advance.

Makeup Tests and Presentation Date Changes

Requests for makeup tests or presentation date changes must be made in advance with the instructor.

Honesty

I am very aware of how easy it is to share your work when it is in electronic form. Be sure you are aware of the Student Code of Conduct found in the [College Syllabus](#). Assisting others with assignments and coding is expected. Providing copies of your work for others to copy is cheating. If you cheat, you fail the course.

How to Get the Most Out of This Learning Experience

Below are simple steps that will make this learning experience even better:

- Take charge of your own learning. Raise questions, prove, explore, go after what you need.
- Be open. Use your imagination, consider new possibilities, and create something new.
- Give as well as receive. Give liberally to co-learners and be prepared to receive a great deal from them.
- Have fun!! Plan to thoroughly enjoy this opportunity to learn and to grow in your professional competence and satisfaction.
- Take advantage of the Raspberry Pi kit and your chance to experiment.

Syllabus Changes:

- The instructor reserves the right to update the syllabus and will inform the class of any changes.

College-wide Syllabus:

- Visit the [College Syllabus](#) available in the main menu to Moodle to view college policies and learning services information.

Tentative Course Itinerary: (subject to change)

The specific day-to-day activities, assignments and topics are located on the course Moodle page.

Week	Topic	Assignments
1	Introductions: Introduction to Course, C Programming, Linux Computer Kit, Coding Tools including Visual Studio Code and GitHub	
2	Variables, Data Types, Operators and Expressions	
3	Data Types, Casting, Input/Output	Sprint 1 Coding Due
4	Control Flow – Branching and Iteration	
5	Nested Loops and Loop Control	
6	Introduction to Arrays	Exam 1, Sprint 2 Coding and Project 1 Due
7	Functions – Arguments and Parameters	
8	Functions – Scoping	
9	Arrays – Initializing and Using,	Sprint 3 Coding and Project 2 Due
10	Multidimensional Arrays	
11	Strings and String Functions	
12	Pointers	Exam 2, Sprint 4 Coding and Project 3 Due
13	Dynamic Memory, File I/O	
14	Structures and Unions, Linked List	
15	Introduction to OOP	Sprint 5 Final Project Due