# CSC 100.01 - Introduction to Programming, Spring 2024 Syllabus

Marcus Birkenkrahe

January 4, 2024

## 1 General Course Information

- Meeting Times: Monday/Wednesday/Friday, 9:00-9:50 AM

- Meeting place: Lyon Building Room 104 (computer lab)

- Professor: Marcus Birkenkrahe

- Office: Derby Science Building 210

- Phone: (870) 307-7254 (office) / (501) 422-4725 (private)

- Office hours: MonWedFri 10-10:50 am & 3-3.50 pm, Tue-Thu 4.15-4.45 pm & by appointment

- Textbook: King (2008). C Programming - A Modern Approach. New York: Norton. Online: knking.com and see also this GitHub repo.

- See also other books used for this course:

  1. Malik DS, C++ Programming: Program Design including Data Structures (8e), Cengage 2018 (Amazon: New from $83.87 - Dec 2023). This is also the textbook for CSC 240 Data Structures with C++ (Fall 2024).
  2. Oualline S, Practical C++ (2e), O'Reilly 2002,
  3. Stroustrup B, Programming - Principles and Practice Using C++, Addison Wesley 2014. See also Author's website.
  4. Haephrasti M and R, Learning C++, Manning 2024.

# 2    Standard and course policies

**Standard Lyon College Policies** are incorporated into this syllabus and can be found at: lyon.edu/standard-course-policies.

The **Assignments and Honor Code** and the **Attendance Policy** are incorporated into this syllabus also and can be found at: tinyurl.com/LyonPolicy.

# 3    Objectives

This course introduces you to programming using C. We cover C++ as an extension. C is a system programming language of pure power: it enables you to converse with the computer at a level unknown to users other high level languages many of which come from C. You also learn about: compilers, working on the command line, text editors Emacs, using C for Internet of Things (IoT) devices, cybersecurity, using pseudocode and process modeling. You get a foundation in critical thinking in concert with two of the three currently most popular languages (cp. TIOBE, 2023): the next three languages are Java, C# and JavaScript, which you can learn in a fortnight after this course). You'll learn some great tools: we use the "hacker's editor" Emacs, the world's foremost software engineering platform, GitHub, and BPMN for process modeling and pseudo code generation. For those interested in computer and data science, this course is a bridge to CSC 240 Data structures in C++ (and Python).

# 4    Student Learning Outcomes

Students who complete CSC 100.01 "Introduction to programming in C++", will be able to:

- Master basic sequential programming skills (conditional statements, loops, functions, input/output, use of data types)

- Explain the basic components of a procedural programming language

- Apply the basics of programming to solve a variety of quantitative problems

- Master computing infrastructure (compiler, editor, shell)

- Research and present a project as a team

- Know how to effectively present assignment results

- Be ready for "Data stuctures with C++" (CSC 240)

- Get ready for "Algorithms" (CSC 265)

- Improve their data literacy skills

# 5 Course requirements / prerequisites

- MTH 110 College algebra or permission from the instructor

- No prior programming or computing knowledge required

- Experience with computers is useful but not critical

- Curiosity is essential, and willingness to try new things!

# 6 Grading system

You should be able to see your current grade at any time using the Canvas gradebook for the course.

| WHEN | DESCRIPTION | IMPACT |
|------|-------------|--------|
| Weekly | Programming assignments | 25% |
| Weekly | Multiple choice tests | 25% |
| Weekly | Participation (class) | 25% |
| TBD | Final exam (optional) | 25% |

Notes:

- To pass: 60% of all available points.

- Participation in class: most of the class will be code-alongs. You have to submit your Emacs Org-mode notebook at the end of the week.

- Tests: weekly online quizzes, which are previewed and reviewed in class.

- Final exam: selection of the most challenging weekly quiz questions.

- You only have to write the final exam if you want to improve your grade at the end of the course. If the final exam result is below your final grade average up to this point, it will be ignored.

- There will be numerous (optional) bonus assignments to improve your midterm and final grades.

# 7 Grading table

This table is used to convert completion rates into letter grades. for the midterm results, letter grades still carry signs, while for the term results, only straight letters are given (by rounding up). This table is also used in Canvas to compute your grades. Transitions between letter grades are rounded (e.g. 89.5% to 90%).

| Percentage | LETTER GRADE* |
|---|---|
| 100% to 89.5% | A (very good) |
| < 89.5% to 79.5% | B (good) |
| < 79.5% to 69.5% | C (satisfactory) |
| < 69.5% to 59.5% | D (passed) |
| < 59.5% to 0% | F (FAILED) |

# 8 Schedule and session content

For **important dates**, see the 2023-2024 Academic Calendar at: catalog.lyon.edu/202324-academic-calendar.

No class on: Monday 19 Feb (Mental-Health Monday) - Monday 18, Wednesday 20, Friday 22 March (Spring break) - Friday 29 March (Easter break). Last day of classes: Wednesday, 8 May. Final exams: 10-15 May.

| WK | ASSIGNMENT | TEXTBOOK CHAPTER | TEST |
|---|---|---|---|
| 1 | Emacs Tutorial | 1 Introducing C | 1 |
| 2 | Program 1 | 2 C Fundamentals | 2 |
| 3 | Program 2 | 3 Input/Output | 3 |
| 4 | Program 3 | 4 Expressions | 4 |
| 5 | Program 4 | 5 Selection | 5 |
| 6 | Program 5 | 6 Loops | 6 |
| 7 | Program 6 | 7 Basic types | 7 |
| 8 | Program 7 | 8 Arrays | 8 |
| 9 | Program 8 | 9 Functions | 9 |
| 10 | Program 9 | 11 Pointers | 10 |
| 11 | Program 10 | 12 Pointers & Arrays | 11 |
| 12 | Program 11 | 13 Strings | 12 |
| 13 | Program 12 | 14 The Preprocessor | 13 |
| 14 | Program 13 | 15 Structs, Unions & Enum | 14 |
| 15 | Program 14 | 16 Advanced Pointers | 15 |
| 16 | Program 15 | From C to C++ | Final |

**Workload:** approx. 4-5 hours per week.

1. Class time = 16 * 3 * 50/60 = 40 hours

2. Tests (home) = 15 * .5 = 7.5 hours

3. Programming assignments: 15 * 1 = 15 hrs (approx)

4. Reading assignments (not graded): 15 * 1 = 15 hrs (approx)