CSC 120 Programming and Problem Solving I Fall 2024

Professor: Blase B. Cindric **Phone:** (330) 829 – 6649 **E-mail:** cindricbb@mountunion.edu
Office: KHIC 048 **Office Hours:** By appointment only

Course Description:

An introduction to object-oriented programming with a focus on algorithms and their use in problem solving. Students will develop concrete problem solving and programming skills through hands-on laboratory experience.

Textbook: There is no textbook for CSC 120 this semester. Selected readings will be assigned throughout the semester from the Mount Union Library's electronic O'Reilly Learning collection, freely accessible by MU students.

Means of Evaluation:

Laboratory Activities	= 70 pts (7.6 % of final grade)
Homework Assignments / Quizzes	= 100 pts (10.9 %)
Programming Projects	= 300 pts (32.6 %)
3 Full-period Exams (100 pts each)	= 300 pts (32.6 %)
Final Exam (cumulative)	= 150 pts (16.3 %)
TOTAL POINTS POSSIBLE:	= 920 pts

Grading Criteria:

93.0% – 100.0%	= A
90.0% – 91.9%	= A–
88.0% – 89.9%	= B+
82.0% – 87.9%	= B
80.0% – 81.9%	= B–
78.0% – 79.9%	= C+
72.0% – 77.9%	= C
70.0% – 71.9%	= C–
68.0% – 69.9%	= D+
62.0% – 67.9%	= D
60.0% – 61.9%	= D–
0.0% – 59.9%	= F

Course Policies:

All assignments are due on or before the beginning of class on the due date, unless specifically stated otherwise. Penalty for late Written, Lab or Programming Assignments = –20% of the points for the assignment per lecture period work is submitted late. No work will be accepted after the last day of classes (December 6, 2024). Make-up work will be available/acceptable only with a legitimate excuse approved by the instructor. (WARNING: Make-up quizzes and tests may be more difficult than regular exams!). Attendance is mandatory for scheduled in-person sessions. A student's grade will be lowered one grade level for missing more than 4 class meetings (excused or unexcused). No cellular phones or other communication devices may be employed during any in-class test, quiz or other evaluation.

Tentative Weekly Schedule: (this course and schedule are subject to change at the discretion of the instructor)

Week	Monday	Wednesday	Friday
1	8/26: Introduction	8/28:	8/30:
2	9/ 2: ** LABOR DAY **	9/ 4:	9/ 6:
3	9/ 9:	9/11:	9/13: [Quiz # 1]
4	9/16: <Program 1 due>	9/18:	9/20: {Written Assgn. 1 due}
5	9/23: [Quiz # 2]	9/25: <Program 2 due>	9/27: **[Exam # 1]**
6	9/30:	10/ 2:	10/ 4:
7	10/ 7:	10/ 9: {Written Assgn. 2 due}	10/11:
8	10/14:	10/16: [Quiz # 3] <Program 3 due>	10/18: ** FALL BREAK **
9	10/21:	10/23: [Quiz # 4] <Program 4 due>	10/25: **[Exam # 2]**
10	10/28:	10/30:	11/ 1: {Written Assgn. 3 due}
11	11/ 4: <Program 5 due>	11/ 6:	11/ 8:
12	11/11: {Written Assgn. 4 due}	11/13:	11/15: <Program 6 due>
13	11/18:	11/20: [Quiz # 5] <Program 7 due>	11/22:
14	11/25: **[Exam # 3]**	11/27: ** THANKSGIVING **	11/29: ** THANKSGIVING **
15	12/ 2:	12/ 4:	12/ 6: <Program 8 due>

CSC 120 Programming and Problem Solving I Fall 2024

Professor: Blase B. Cindric **Phone:** (330) 829 – 6649 **E-mail:** cindricbb@mountunion.edu

Accessibility Support Statement:

The University of Mount Union values disability as an important aspect of diversity and is committed to providing equitable access to learning opportunities for all students. Student Accessibility Services (SAS) is the campus office that collaborates with students who have disabilities to provide and/or arrange reasonable accommodations based upon appropriate documentation, nature of the request, and feasibility. If you have, or think you have, a temporary or permanent disability and/or medical diagnosis in any area such as, physical or mental health, attention, learning, chronic health, or sensory, please contact SAS. The SAS office will confidentially discuss your needs, review your documentation, and determine your eligibility for reasonable accommodations. Accommodations are not retroactive, and the instructor is under no obligation to provide accommodations if a student does not request accommodation or provide documentation. Students should contact SAS to request accommodations and should discuss their accommodations with their instructor as early as possible in the semester. You may contact the SAS office at (330) 823-7372; or via e-mail at studentaccessibility@mountunion.edu.

Course Goals:

Students will learn to create algorithms to solve programming problems as well as learn to write Java programs to implement algorithms.

Course Objectives:

Upon completion of this course, students should be able to:

- create, compile and execute Java applications.
- declare, instantiate and use objects that come from one or more class definitions.
- declare and use constants and variables represented as one of the Java primitive data types.
- write a syntactically-correct Java method signature, when given the number and types of parameters and the return type.
- write a complete, syntactically-correct Java method to solve a specific problem.
- write a syntactically-correct Java method invocation, when given the signature of the method.
- write a complete, syntactically-correct Java class to model graphical or non-graphical entities.
- write syntactically- and semantically-correct Java statements to compute the total, average, minimum and maximum value of a series or collection of numerical values.
- given the source code for a Java program that manipulates objects from one or more of: the Graphics class, GUI component classes, Strings, one- or two-dimensional arrays, and/or user-defined classes (graphical or non-graphical), and that outputs values before, during and after manipulation, write the output of the program on paper.
- given a programming problem whose solution requires using objects from one or more of: the Graphics class, GUI component classes, Strings, one- or two-dimensional arrays, and/or user-defined classes (graphical or non-graphical), write an algorithm to solve the problem.
- given an algorithm that uses objects from one or more of: the Graphics class, GUI component classes, Strings, one- or two-dimensional arrays, and/or user-defined classes (graphical or non-graphical), translate the algorithm into syntactically- and semantically-correct Java source code.

CSC 120 Programming and Problem Solving I Fall 2024

Professor: Blase B. Cindric **Phone:** (330) 829 – 6649 **E-mail:** cindricbb@mountunion.edu

Expectations of Students:

CSC 120 includes the expectation that students will spend a significant amount of outside-of-class time working on assignments. Computer programming assignments must be completed outside of class time, your instructor may give written homework assignments to be completed between class sessions, and in-class labs that are not finished during the class period when assigned must be completed by 5:00 pm on the FIRST class day after the date of the lab to receive full credit. You simply cannot succeed in this course without spending time outside of class studying and working on course assignments / activities.

Students are advised that it is generally accepted in the computer education community that the best way to learn computer programming techniques is PRACTICE, PRACTICE and MORE PRACTICE! Writing lots of programs and code will pay off with learned abilities and skills. You should expect to spend five to six hours a week outside of class working on CSC 120 activities.

Some students will have no prior programming experience at the start of this course, while others may have written some computer programs in the past. No prior programming experience is assumed on the part of students, but the course does move at a brisk pace. Students are advised to keep up with assignment deadlines and course work throughout the semester, and if more challenge is desired, to explore additional programming techniques as appropriate.

Academic Honesty / Use of AI:

Academic Integrity is at the heart of the mission and values of the University and is an expectation of all students. Maintaining academic integrity is a means of obtaining grades that accurately reflect student learning and understanding of the course material. All work submitted by you in this course MUST be **YOUR** work. Please refer to the *Academic Honesty* section in the current Undergraduate Catalog.

It is OK or even encouraged to help each other with labs, but you should attempt to figure things out yourself first.

For Programming Project assignments you are each to do your own work. When you need help, the first resource should be your notes and samples from class and your other labs and projects. Next should be one of the CS professors or the teaching assistant(s).

If you do ask a classmate or another student for help, you must document explicitly in that section of code and in the comment header at the top the name of the person who helped and what help they provided (which method and/or code segment). If you use code fragments you found in a book, on a website, or from some other source, you must document explicitly in that section of code and in the comment header at the top the exact source and exactly which code segment you copied.

You must never copy other students' code. You should make no attempt to even look at their code unless they have asked you for help.

Use of Chat-GPT or other AI-based content generators is strongly discouraged, and is prohibited for work you are submitting as your own for grading. The main goal of this course is to teach you computer programming and problem-solving skills, and you need to solve the problems yourself in order to learn this content. Many times, the solutions given by AI tools use a completely-different programming style than we use in this course, and it will be easy to see that you didn't write the code yourself; you had someone else (either another human or an AI) write it. Either case represents plagiarism for the assignment in question. The University of Mount Union Catalog clearly states that, "Plagiarism and/or any other form of cheating or dishonesty will subject the student involved to possible suspension or dismissal from the University." A majority of the points that make up your grade for the course come from in-class evaluations/tests when you cannot use an AI tool, so you are hurting yourself if you use one for out-of-class work.