CSC 120 Programming and Problem Solving I Fall 2021

Professors: Louise Campbell (Section 01) Blase Cindric (Section 02)

Course Web Page: See CSC 120 D2L Course Home Page

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 SAFARI TECH BOOKS collection, freely accessible by MU students.

Means of Evaluation:

Laboratory Activities ..... =	70 pts	(7% of final grade)
Homework Assignments / Quizzes ..... =	100 pts	(10%)
Programming Projects ..... =	300 pts	(30%)
3 Full-period Exams (100 pts each) ..... =	300 pts	(30%)
2 Lab Practica (on-computer tests) ..... =	80 pts	(8%)
Final Exam (cumulative) ..... =	150 pts	(15%)
TOTAL POINTS POSSIBLE: ..... =	1000 pts	

Grading Criteria:

92.5% – 100.0%	=	A
89.5% – 93.4%	=	A–
86.5% – 89.4%	=	B+
82.5% – 86.4%	=	B
79.5% – 82.4%	=	B–
76.5% – 79.4%	=	C+
72.5% – 76.4%	=	C
69.5% – 72.4%	=	C–
66.5% – 69.4%	=	D+
62.5% – 66.4%	=	D
59.5% – 62.4%	=	D–
0.0% – 59.4%	=	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 Programming Assignments = –20% of the points for the assignment per lecture period work is submitted late. No work is accepted after the last day of classes. 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 both lecture and lab 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:

Week	Monday	Wednesday	Friday
1	8/23: Introduction	8/25:	8/27:
2	8/30:	9/ 1:	9/ 3:
3	9/ 6: ** *LABOR DAY* **	9/ 8:	9/10: [Quiz # 1]
4	9/13: <Program 1 due>	9/15: {Written Assgn. 1 due}	9/17:
5	9/20: [Quiz # 2]	9/22: <Program 2 due>	9/24: **[Exam # 1]**
6	9/27:	9/29:	10/ 1:
7	10/ 4:	10/ 6: (Practicum 1)	10/ 8: {Written Assgn. 2 due}
8	10/11:	10/13: <Program 3 due>	10/15: ** *FALL BREAK* **
9	10/19: [Quiz # 3]	10/20: [Quiz # 4]	10/22: <Program 4 due>
10	10/25: **[Exam # 2]**	10/27:	10/29:
11	11/ 1: {Written Assgn. 3 due}	11/ 3: <Program 5 due>	11/ 5:
12	11/ 8:	11/10: {Written Assgn. 4 due}	11/12: <Program 6 due>
13	11/15:	11/17: (Practicum 2)	11/19: [Quiz # 5]
14	11/22: <Program 7 due>	11/24: ** *THANKSGIVING* **	11/26: ** *THANKSGIVING* **
15	11/29:	12/ 1: **[Exam # 3]**	12/ 3:
16	12/ 6:	12/ 8:	12/10: <Program 8 due>

CSC 120 Programming and Problem Solving I Fall 2021

Professors: Louise Campbell (Section 01) Blase Cindric (Section 02)

Accessibility Support Statement:

Any student with a documented disability requiring academic accommodations is requested to speak with the Director of Student Accessibility Services (330 – 823 – 7372) and the instructor, as early in the semester as possible. All discussions will remain confidential.

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:

- explain how software components interact to execute Java programs and manage data
- 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 noln-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 2021

Professors: Louise Campbell (Section 01) Blase Cindric (Section 02)

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 the start of the SECOND class session 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:

Students should be familiar with the university-wide policy concerning academic honesty in the university catalogue and referenced in the student handbook.

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

The University of Mount Union Catalogue clearly states that, "Plagiarism and/or any other form of cheating or dishonesty will subject the student involved to punitive action ranging from failure of an assignment to possible suspension or dismissal from the University."