Augusta University ▪ School of Computer and Cyber Sciences

Principles of Computer Programming I
CSCI 1301 C/D
Fall 2019

Instructor: Neea Rusch
Email: nrusch@augusta.edu

Course Dates: August 14, 2019 – December 12, 2019
Final Exam Date: December 12, 2019 (Thursday, 5:00 – 7:00 pm)

Lecture: Allgood Hall (AH) E152
Mondays 5:30 – 6:45 pm
Wednesdays 5:30 – 6:45 pm

Lab: Allgood Hall (AH) E365
Mondays 7:00 – 8:50 pm (Section C)
Wednesdays 7:00 – 8:50 pm (Section D)

Course Description

A rigorous study of the principles of computer programming with emphasis on problem solving methods which result in correct, well-structured programs. Other topics: an introduction to data representation, data types and control structures, functions, and structured data types.

Learning Outcomes

Students who successfully complete this course will be able to:

1. Perform standard program Input and program Output using the keyboard and the monitor
2. Declare and use user-defined variables, and constants using the appropriate data types
3. Declare, define, and call user-defined functions
4. Write and evaluate expressions using arithmetic, relational and logical operators
5. Control the flow of program execution using the appropriate sequential, selection, and repetition statements
6. Define, create and manipulate arrays
7. Understand and implement classes and objects

By the end of this course, students will be able to solve computer-programming challenges using correct, well-structured applications written in the C# programming language. Students will have a basic knowledge and comfort using control structures, object-oriented design, classes, methods, properties, data types, and data structures. The student also will develop computational thinking skills and practices.

Format and Procedures
CSCI 1301 is an academically rigorous four (4) credit hour course consisting of lecture and laboratory portions: both are required to succeed. This class is an on-campus class. Lectures are devoted to general explanations of the concepts and ideas underlying the topic at stake. Laboratory will be devoted to hands-on practice and experiments.

The initial lab exercises assume no previous programming experience. Exercises increase in complexity and level of challenge as the course progresses. All programming exercises are designed to complement the lecture material.

Homework assignments will assist the students in making sure they understand classes expectations and the content of the lecture, as well as to practice their coding and problem-solving skills. The progression of the students will be regularly tested and assessed through quizzes, projects and tests. Active and relevant participation during the lectures and laboratory sessions is appreciated.

Student Expectations
- Read this entire syllabus carefully
- Participate actively in all class discussions
- Complete homework assignments: read your notes before starting the homework assignment, make sure you understand it completely before considering it done
- Work through each lab and make sure you understand the theoretical concepts
- Come prepared and on time to classes, exams and quizzes

General Class Rules
- Attendance is not mandatory; however, you are strongly encouraged to attend class. Come to class on time and stay until the end of the lecture: late arrival and early departure disturb the learning experience for everyone.
- You are responsible for all course material, and your decision to attend lectures, do the assigned reading and coursework. I do not repeat lectures or provide notes for those who miss class.
- It is the student's responsibility to initiate a withdrawal before midterm, but I reserve the right to withdraw a student that missed 10% of class time and half of the quizzes and tests.
- A student not withdrawn from a course who stops attending class is subject to receiving a grade of WF or F.
- Come to your section's laboratory. If you want to change your section, find a fellow student willing to switch with you and go to the registrar's office.
- Quiet chat and mutual help are acceptable, sharing solutions is forbidden. You may verbally discuss your general approach and solution strategy. Do not share files, show your code, or dictate what to type.
- All coursework is individual coursework. Identical or similar programs turned in by two or more students receive a grade of zero. Violations of the AU Academic Honesty Policy will be reported.

Grading
Students will be evaluated using four different types of evaluation:

1. Homework assignments will be given during the semester: they are not expected to be handed back, and won't be graded, but quizzes (closed book and timed (± 10 min.)) with questions taken or inspired from those assignments and the lab will be given.
2. Projects will be carried at home or during laboratory.
3. There will be in-class exams, held during the regular class periods.
4. The final exam will take place during the exam period.

Your grade will be computed as follows:

Quizzes (×5)	10%
Projects (×3)	10%
In-class Tests (×2)	40%
Final Exam	40%

Course Grade Scale

A	**B**	**C**	**D**	**F**
90 – 100 %	80 – 89 %	70 – 79 %	65 – 70 %	Below 65%

I do not curve individual examinations. At the end of the course, the class average is calculated to determine if an overall scaling of grades is necessary.

Exam Absence
No makeup quizzes or exam will be allowed. Exam absences must be coordinated with me prior to the exam. Under certain circumstances and with prior permission, I may grant you permission to count your Final Exam grade as a missed exam grade. Note that this allowance is available only with my prior permission and is only available to replace one missed exam. Unexcused missed exams will result in a grade of zero (0). Any student missing the final exam without a documented excuse (brought to me or to the dean of Student Life), or who has not taken action to withdraw, will receive a grade of F. In case of a documented emergency at the time of the final, the student may be allowed to receive a grade of I.

Hardware & Software Requirements
For this class, you will need to access a computer with Visual Studio installed on it. You can either:

- Visit one of the Computer Labs that are accessible to every student,
- Use the lab reserved for students enrolled in a CSCI / AIST / MS-IMS class, in University Hall, room 131,
- Use your personal computer. Instructions on how to install and configure the software will be given during lab, download codes will be accessible from https://www.augusta.edu/its/software.php or using your onthehub account.

Supplemental Reading
Deitel & Deitel, *Visual C# How to Program*, 6th edition, © 2017; Pearson; ISBN-13: 9780134601540. This textbook is optional, and can be accessed online at: https://www.safaribooksonline.com/library/view/visual-c-how/9780134628820/

Academic Accommodations & Assistance
I should be your first point of contact for any questions regarding the content of this class, but many other resources are available:

- For tutoring resources, consult Academic Success Center (or "ASC"). Tutoring is available for Computer Science on the first floor of University Hall. You can schedule appointments at https://augusta.campus.eab.com/.
- Testing & Disability Services can help accommodate this class. Contact Testing and Disability Services (Galloway Hall; 706.737.1469; www.augusta.edu/tds/) as soon as possible for more information and/or to initiate the process for accessing academic accommodations.
- Student Counseling & Psychological Services ("SCAPS") is here to assist students with a variety of personal, developmental, and mental health concerns.
- Student Assistants – we will have "embedded tutors" this semester.

Preliminary Course Schedule
This schedule is subject to change, but provides an indication of the pace, assignments, and major deadlines that you will need to plan for the semester.

Week	Date	Notes	Topic
1	08/14	First day of class	Syllabus, Introduction
2	08/19	-	Reserved Words, Variables, Data types
3	08/26	08/26: Quiz 1	Operations, Casting, Reading from User
4	09/02	09/02: no class	Intro to Object-Oriented Language
5	09/09	09/09: Quiz 2	OOP (cont.)
6	09/16	09/16: Project 1 09/18: Exam 1	Review Session
7	09/23	-	Control Structures and if Statements
8	09/30	-	switch Statement
9	10/07	10/07: Quiz 3	while Loops, validating input
10	10/14		do-while, char, Random
11	10/21	10/21: Quiz 4 10/27: Project 2	char (cont.)
12	10/28	10/30: Exam 2	Review Session
13	11/04	-	Arrays, for Loops
14	11/11	-	Arrays, foreach
15	11/18	11/18: Quiz 5	OOP Recap, static, class diagrams
16	11/25	Project 3 11/27: no class	theory wrap-up
17	12/02	12/04: last day of class	Final Exam Review
18	12/12	**Thursday** 5:00 – 7:00 pm	Final Exam