Augusta University ▪ School of Computer and Cyber Sciences

Principles of Computer Programming I
CSCI 1301 C/D
Spring 2020

Instructor: Neea Rusch
Email: nrusch@augusta.edu

Course Dates: January 6, 2020 – May 6, 2020
Final Exam Date: May 6, 2020 (Wednesday, 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 consist of discussion focused on concepts and principles of computer programming. 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 students in making sure they understand class 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 and tests. Active participation during the lectures and laboratory sessions is expected.

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 lectures, exams and quizzes

General Class Rules

- Attendance is not mandatory; however, you are strongly encouraged to attend every lecture and lab. 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 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 to a fellow student, 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.
- Any violations of the AU Academic Honesty Policy will be reported.

Grading

Students will be evaluated using 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. Quizzes (closed book and timed 10-15 min.) with questions taken or inspired from homework and lab assignments will be given.
2. Code challenges will be given weekly and are to be completed independently on D2L.
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:

Lab attendance	5 %
Code Challenges	5 %
Quizzes (×5)	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

There will be no makeup quizzes or exams. 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 zero grade. 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 F. In case of a documented emergency at the time of the final, the student may be allowed to receive a grade I.

Hardware & Software Requirements

For this class, you will need to access a computer with Visual Studio installed on it. Options:

- Use your personal computer
- Visit one of the Computer Labs that are accessible to every student,
- Use the lab reserved for students enrolled in CSCI / AIST / MS-IMS class, in University Hall, room 131

Instructions on how to install and configure the software will be given during the first 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 am 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/) 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 – Tim Cuny will be the course embedded tutor throughout the 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	01/06	First day of class	Syllabus, Introduction First program, displaying output
2	01/13	-	Variables, data types, operators
3	01/20	01/20: no class 01/22: Quiz 1	Casting; converting, reading input
4	01/27		Control Structures, if Statements
5	02/03	02/03: Quiz 2	if (cont.), switch Statement
6	02/10	02/12: Exam 1	Review Session
7	02/17	-	while Loops,
8	02/24	-	do-while, input validation
9	03/02	03/02: Quiz 3	Arrays, for loops, foreach
10	03/09	03/11: Quiz 4	Arrays (cont.), char, Random
11	03/16	no classes	-
12	03/23	no classes	-
13	03/30	04/01: Exam 2	Review Session
14	04/06	Spring break: no classes	-
15	04/13	Remote learning	OOP: get, set, methods, constructors,
16	04/20	Remote learning	ToString, static keyword, class diagrams
17	04/27	04/29: Quiz 5 due 04/29: Last day of classes	Review
18	05/06		Final Exam