Mr. Ryan Darras rdarras@uccs.edu C: 719.322.8047

UCCS CS3020-001; Spring 2019 Advanced Object Technology using C#/.Net

Meeting Time & Location

Days: Mondays & Wednesdays
Time: 3:05pm – 4:20pm
Location: ENGR 109
Office hours: MoWe 4:20pm – 5:30pm

Note on office hours: As a graduate student at UCCS, I have no permanent office, and I may or may not be on campus outside of the hours I have provided. The best way to schedule time with me outside of class is to email to make arrangements.

Instructor

Ryan Darras
rdarras@uccs.edu
C: (719) 322-8047

Course Catalog Description

C# class construction principles, delegates, threads, event handling, GUI components, observer pattern, standard collections, generic parameters, enumerators, custom components, UML representation, abstract classes, interfaces, object persistence, remoting, and refactoring. ***Prer., CS 1450 or GDD 2200.***

Textbook

Fundamentals of Computer Programming with C#
Svetlin Nakov, Veselin Kolev & Co.
ISBN 978-954-400-773-7

Note on textbook: The textbook is available for free online as a supplementary resource. I understand that many students don’t learn effectively by reading so I will provide alternative resources throughout the semester to videos, blogs, and other online media.

Software & Tools

Visual Studio 2017 (Community Edition)
https://visualstudio.microsoft.com/downloads/
In the installer, make sure the following boxes are checked:

- .NET desktop development
- Universal Windows Platform development
- Mobile development with .NET

GitHub
https://github.com/
Create an account (I recommend NOT using your UCCS email address. This is an online repository that is often requested in job applications and you will want access to it after you graduate).

Interfacing with GitHub:

GitHub itself is an online repository, in which you need some sort of tool to interface with. Below I recommend three options.

SourceTree (This is what I use)
https://www.sourcetreeapp.com/

GitHub Desktop
http://desktop.github.com

Command Line
https://gitforwindows.org/

Grading

All grades are based on a scale from 0-100 as follows:

	93-100 A	90-93 A-
87-90 B+	83-87 B	80-83 B-
77-80 C+	73-77 C	70-73 C-
67-70 D+	63-67 D	60-63 D-
0-60 F		

I believe that software development and computer science is best learned through implementation. Therefore, this class will be heavily weighted on assignments and homework vs quizzes and exams. The following table shows the class weighting scheme.

Type	Each	Total	Weight
Homework (~12)	~50	600	60%
Midterm	150	150	15%
Final	150	150	15%
Final Project	100	100	10%
Total		1000	100%

Note that a linear shift may be applied to **final** grade averages as a one-time scale at the instructor's discretion.

Course Schedule

This schedule represents a reasonable expectation as to the pace we will address topics throughout the semester. I reserve the right to amend this syllabus at any time. **Final deadlines, exam dates, and other announcements will always be posted on canvas and are to be considered over dates on this syllabus.**

Week	Resource	Topics	Assign. Due
1/21-25	Chap 1-9	C# syntax, .NET, Visual Studio, GitHub	
1/28-2/1	Chap 1-9	C# syntax, .NET, Visual Studio, GitHub	HW 1
2/4-8	Chap 11,14,20	Object Oriented Programing (OOP)	HW 2
2/11-15	Chap 14	Generics and Interfaces	HW 3
2/18-22	Chap 22	Lambda Expressions and LINQ	HW 4
2/25-3/1	Chap 22	Events, Delegates, and Anonymous Methods	HW 5
3/4-3/8	Chap 14,18	Dictionaries and Enumerations	HW 6
3/11-15	Chap 3,20	Singletons and Conditional/Null-Coalescing Operators	HW 7
3/18-22	See Canvas	Midterm prep & Midterm	
3/25-29		Spring Break	
4/1-5	See Canvas	Threading	HW 8
4/8-12	Chap 10	Recursion	HW 9
4/15-19	See Canvas	WinForms	HW 10
4/22-26	Chap 15, See Canvas	File IO and Serialization	HW 11
4/29-5/3	Chap 22	Extension Methods	HW 12
5/6-10	See Canvas	Final Review Week	Final Project
5/13-17	See Canvas	Finals week	

Programming Assignments

There will be between 10-12 programming assignments. Programming assignments are to be completed on your own. However, you may discuss any component of the assignment with your classmates and help each other to debug, but there cannot be a physical or electronic record of your conversation (no paper, files, disks, or code of any form) taken away from the conversation. While you are encouraged to help each other debug, you may not write or dictate any portion of the code. You must write your own code. This includes taking code from the internet; you may not use any code that you have found on the internet within your work. Copying any portion of the code will result in an automatic zero for the project for all students involved. Two or more instances of this in the course will result in an automatic failure for the course. I expect most assignments to take about 6 hours per week to complete.

Assignments are due at the beginning of class on the assigned day. Your work must compile and run, be fully commented, meet all standards for style and documentation, and meet the file-naming requirements for the assignment. To turn in your assignment, make sure your project has been pushed to your repository and submit the post-assignment documentation to Canvas in the appropriate section.

Project

Over the course of the semester you are to develop a project in which you decide the parameters and requirements. This project can simplify or solve real problems, it can be a game or a tool, or it can simply be complex enough to consider interesting. I expect this project to take anywhere from 10 to 30 hours to complete based on the problem, solution, and implementation. Think of your ideas early in the semester and find a chance to run it by me, as I will be able to give you some feedback that will help you determine if your idea meets the complexity requirement.

Late Work

Work not submitted prior to the deadline will be given a zero barring extenuating circumstances. If an emergency arises and you are unable to finish your assignment you must notify me as early as possible. This caveat goes both ways; you can expect to have every assignment graded within a week of its due date.

Class Attendance and Participation

This is a 3000-level course; therefore, I expect every student to act responsibly. If you are unable to attend a class period, it is your responsibility to recover any information that you missed on your own time. “I didn’t know because I wasn’t there” is not a valid excuse for a missing assignment, missed exam, etc. You are 100% responsible for all material and announcements covered in class.

Exams

Exams are in place as a measurement tool for both the students and the instructor. Exams may take the form of a traditional, in-class written exam or in-class lab practica where you will have a set period of time to develop a solution to a problem. A take-home final exam may also be considered.

Plagiarism & Cheating

Absolutely no cheating, copying, or plagiarizing on the exams or assignments will be tolerated. Students are encouraged to discuss general concepts individually and in class; however, each student is expected to develop their own assignment and exam solutions. Cheating will result in an AUTOMATIC ZERO (0) for the entire exam or assignment in question. The second incident will result in failure for the course. For further details on academic honesty the student is referred to the University Catalog.

Dropping the Course

Dropping of a class after the deadline listed in the class schedule is governed by departmental and college policy. The student must show documented evidence supporting reasons for a request to drop a class after the deadline. Each request is considered on an individual basis for determining acceptance.

The last day to drop Spring 2019 full semester length Main Campus courses within the myUCCS Poral ***without*** dean or instructor signatures is April 5th.

NOTE: Full semester length courses dropped after Census Date (February 6) within your myUCCS Portal are non-refundable and are automatically assigned a grade of W (Withdrawn).