

ADV APPLICATION DEVELOPMENT WITH C#

ISAM 5430.01: CLASS 10604. SPRING OF 2020.

MONDAY, 1:00 – 3:50 PM @ SSB 2311 (LECTURES) → SSB 2.201.03 (LAB).

1. Instructional Information	1
2. General Information	1
3. Learning Outcomes	2
4. Course Criteria	2
5. University Policies	4
6. Course Schedule	5

Disclaimer: The information contained in this class syllabus is subject to change without notice. You are expected to be aware of any additional course policies presented by the instructor during the course.

1. INSTRUCTIONAL INFORMATION

Designation	Information	Location & Time
Michael Yu-Chi Wu, Ph.D. Instructor	wum@uhcl.edu 281-283-3169 The instructor teaches the class and facilitates your learning outside the class. The instructor may be using one of the MIS labs to help students.	SSCB 3.202.02 , MIS Department Monday, 10:00 AM – 1:00 PM; Tuesday, 10:00 AM – 1:00 PM. <i>Other times by appointment</i> (send an email first).
Mo, Siu Hang Teaching Assistant	The goal of the TA is to facilitate students learning outside the class.	You can find the schedule and location of the TA and other TA's on http://mis.uhcl.edu .
Tutors	MIS Tutoring may be available for this course. Phone: (281) 283-2450 <i>For more info, see Section 4.6.</i>	3 rd floor of the Student Success Building (SSB) www.uhcl.edu/studentsuccesscenter
Online Resources	<i>SharePoint</i> {documents and notes} <i>Blackboard</i> {submissions and grades} <i>Github classroom</i> {assignments}	http://c.mikeywu.com/ https://blackboard.uhcl.edu/ http://git.mikeywu.com/

2. GENERAL INFORMATION

2.1. DESCRIPTION

The course focuses on building high-performance business applications using the C# ("C-sharp") programming language in the .NET 4.6 environment. It begins by developing a very strong foundation in coding—which includes proper understandings and usages of data types, variables, methods / functions, loops, conditions, and controls—with C#. Additionally, the course will delve into basic application development in the .NET environment.

3. LEARNING OUTCOMES

Upon the completion of the course, the course allows students to...

- 1) apply the C# language and the .NET framework to implement software solutions;
- 2) utilize appropriate C# data types, such as int, bool, double/float, decimal, char, string, and arrays;
- 3) practice various techniques and algorithms that involve tracking, mapping, permutation, and recursion;
- 4) develop software codes that make use of variables and control statements with loops and if-statements;
- 5) employ data structures—lists, arrays, and hash tables/dictionaries—to solve new problems;
- 6) organize C# codes into classes, fields, properties, and methods;
- 7) employ object-oriented principles—encapsulation, abstraction, inheritance—into codes;
- 8) interpret written requirements into C# codes involving programming classes, operations, and attributes;
- 9) tabulate test cases, use cases, and runtime data;
- 10) identify compilation errors, runtime errors, and logical errors in the codes;
- 11) operate Visual Studio.NET as an integrated development environment to create and build C# solutions
- 12) experiment various Visual Studio features, such as intellisense, debugging, error logs, and variable watches;

3.1. PRE-REQUISITES

Some programming background or **ISAM 5030: Introduction to Business Application Programming**.

3.2. COURSE FORMAT

In general, all the classes will consist of mostly lectures of concepts, logic, data structures, and syntaxes with some hands-on workshops and lab work to help engage students to move in proper directions. As there are many different ways to implement programming problems, the instructor will initially give lectures on how to solve these programming problems. Short workshops will then follow these lectures to help students to start developing these codes. Nevertheless, due to limited class-times, students will then complete any unfinished codes done during these workshops at home.

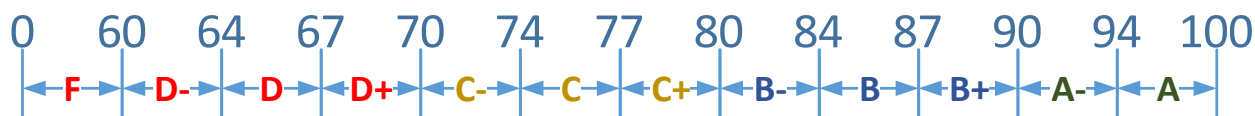
As a graduate student, you are expected to spend between three and five times the credit hours outside of the class per week by completing your own codes and assignments. That is, you are expected to spend between 9 and 15 hours every week at home. Completing your codes once is never enough. There are two factors that determine whether you're a good coder or not. First, you must be able to code quickly. Second, you must be able to make less mistakes. Once you have achieved these goals, you are good to go. If not, you will need to spend more time practicing at home.

3.3. TEXTBOOK AND OTHER RESOURCES

- 1) Deitel, P., & Deitel, H. (2016). *Visual C#, How to Program*, Sixth Edition. Upper Saddle River, NJ, USA: Pearson Education, Inc.
- 2) Students will need to use a computer with Windows and Visual Studio 2017/2019 installed. Alternatively, students can work on these computers located in any one of the computer labs. Additional course materials can be found on the Internet and on the Blackboard; see Section 1 (Instructional Information) for the hyperlinks.

4. COURSE CRITERIA

4.1. METHODS OF EVALUATION AND GRADING



Item	Weight	Item	Weight
Assignments	15%	*Coding Test 1	20%
Quizzes & Participations	20%	Coding Test 2	25%
		Coding Test 3	20%
		Total	100%

** Coding Test 1 takes the highest score of Coding Test 1 (A) and Coding Test 1 (B): see Course Schedule in Section 6 on page 5 for more information.*

The grading scale indicates the range of percentages that the letter grade corresponds. Additionally, a grade of **incomplete** is not available in this course. The percentage weight for each of the gradable items is listed in the table.

4.2. ASSIGNMENTS

Students must practice writing codes at home many times by applying their thinking processes to solve various problems. The purpose of homework assignments is to give students the opportunity to explore coding at home. In addition, homework assignments can help students prepare for all the coding tests and quizzes, which can validate whether they have practiced enough at home or not.

Before working on any assignment, students will need to review all the lecture notes, presentations, and workshops at home, in which they will be redoing all the codes taught in the class by going over each part individually. Then, students can attempt writing codes in the assignment. This way, students can explore various ways of how to write codes on their own.

Students will be receiving a GITHUB link to an assignment to accept these assignments, whereby creating their own git repositories before working on the assignment. The first step is to clone the git repository created into Visual Studio. They will then create a solution in Visual Studio within the same git repository. Afterward, they will start developing codes for their assignments. When students have completed the assignment, students must first commit all their codes to the local git repository and then push all their changes to the github server. This way, the TA and the instructor will be able to look at their assignments.

Submitting an assignment to blackboard, emails, or any other form are not acceptable. That is, students who are stuck with git must seek help from the TA and the instructor who can show them how to do so correctly. The **deadline** of each homework assignment pushed to the github server is **before the class starts** (by 1PM).

4.3. QUIZZES & PARTICIPATIONS

The course will include many quizzes. A quiz can be administered at any time during the class. Each quiz will contain a coding problem. The coding problem is based on the lectures and homework assignments from the preceding classes. Homework assignments, lectures, and the “Self-Review Exercises” located in the back of each chapter will help you prepare for the quiz. Each quiz is closed book, closed notes, closed for discussions, and to be completed individually on either on paper or a computer depending on the nature of the problems.

Sometimes, the instructor may require the student to perform specific class work or participate a task. These results may be account for participations.

4.4. CODING TESTS

Rather than having a midterm exam and a final exam, the course will have four coding tests throughout the semester, held outside the class times. These tests are computer-based and will require students to develop C# in Visual Studio.NET on the computers. The goal of these tests is to assess general web designing skills. Section 6 (Course Schedule) summarizes the coverages of all the computer-based tests.

One shall note that the first coding test, both Coding Tests 1 (A) and 1 (B), involves algorithms, taking the second coding test 1 (Coding Test 1 (B)) is optional, as the highest score of both tests will be used to compute the final score of the coding test 1. Therefore, if a student is satisfied his/her Coding Test 1 (A) score (e.g. 90% or above), he/she does not need to take Coding Test 1 (B).

Coding tests are **closed-notes** and **closed-book** tests. These tests will require students to complete developing codes during the entire class in the beginning of the class before the lecture starts. In addition, the Internet will be disabled in every test. The instructor will then collect solution files from all the students on a USB flash drive or an SD card. Thus, every student is responsible to provide accurate and correct solution files.

To evaluate students’ performances as accurately as possible, the course requires students to complete computer-based coding tests. After the students have completed a test, the instructor will create all the rubrics based on the learning outcomes of the test. The weights of these rubrics are determined by several factors: the difficulty level of the item (i.e. logics) and the performances of all the students (i.e. curves). The instructor then assigns a percentage rating to each item as described in the table.

Rating	What it means
0%	The feature is unimplemented, or the implemented feature does not meet any requirement.
50%	The implemented feature partially satisfies the requirements.
100%	Codes fully satisfy the requirement with no mistakes or very negligible mistakes.

4.5. MAKEUPS

Opportunities to make up missed quizzes, tests, and final exam are only granted for cases of extreme and unusual circumstances, as judged by the instructor. That is, students may be asked to provide a physician note. Otherwise, missed quizzes, tests, and final exams are assigned a grade of zero. The content and format of makeups, if given, differ from normally scheduled exams.

4.6. TUTORING SERVICES

The Student Success Center is a comprehensive academic support resource for the UHCL student community. The Center's services are free of charge and include tutoring for this course, study skill development and academic coaching. You can visit the Student Success Center webpage at www.uhcl.edu/studentsuccesscenter or call 281-283-2450 to preview our services and set appointments. The Student Success Center is located on the 3rd floor of the Student Success Building (SSB)."

5. UNIVERSITY POLICIES

5.1. LAST DAY TO WITHDRAW

The last day to withdraw from this course without grade penalty is **April 14, 2020**. Withdrawals from the course must be filed in the Office of Enrollment Services. Students who stop attending the class without officially dropping the course will receive a failing grade in the course.

5.2. STUDENT LIFE POLICIES

The Student Life Policies can be found by going to <http://www.uhcl.edu/student-services> and selecting "Student Life Policies."

5.3. STATEMENT ON ASSESSMENT

Guidelines in the School of Business state that:

"The School of Business may use assessment tools in this course and other courses for curriculum evaluation. Educational Assessment is defined as the systematic collection, interpretation, and use of information about student characteristics, educational environments, learning outcomes and client satisfaction to improve program effectiveness, student performance and professional success.

This assessment will be related to the learning objectives for each course and individual student performance will be disaggregated relative to these objectives. This disaggregated analysis will not impact student grades, but will provide faculty with detailed information that will be used to improve courses, curriculum, and students' performance."

5.4. ACADEMIC HONESTY POLICY

The Academic Honesty Policy at UHCL (found on the Dean of Students' website, the Faculty Handbook, the Student Handbook, the Senior Vice President and Provost's website, the Graduate Catalog, and the Undergraduate Catalog) states: Academic honesty is the cornerstone of the academic integrity of the university. It is the foundation upon which the student builds personal integrity and establishes a standard of personal behavior. Because honesty and integrity are such important factors in the professional community, you should be aware that failure to perform within the bounds of these ethical standards is sufficient grounds to receive a grade of "F" in this course and be recommended for suspension from UHCL. The Honesty Code of UHCL states "I will be honest in all my academic activities and will not tolerate dishonesty."

5.5. SPECIAL ACADEMIC ACCOMMODATIONS (AS SPECIFIED BY THE AMERICANS WITH DISABILITIES ACT)

If you believe you have a disability requiring an accommodation, contact Disability Services at 281-283-2648 or disability@uhcl.edu as soon as possible and complete their registration process. The University of Houston System complies with Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act of 1990, pertaining to the provision of reasonable academic adjustments/auxiliary aids for students with a disability. In accordance with Section 504 and ADA guidelines, each University within the System strives to provide reasonable academic adjustments/auxiliary aids to students who request and require them.

6. COURSE SCHEDULE

Please study the assigned chapters and sections in the textbook before the class starts. A **quiz** can occur in any week and anytime of the class. Weekly assignments are due before the class starts (e.g. 1 PM)

Classes		Objectives	Expectations	
01	1/27	Introductions Introduction to Computers • Visual Studio • Write C# Codes in console app • Input and output data • Variables • String interpolations • operators • Boolean operators and if statements • <i>github.com</i>	Deitel Deitel	Chapter 1 Chapter 3
02	2/3	Algorithm Development and Control Statements (Part 1) Selection statements • Iteration Statements • Break/Continue • Logical Expressions • Short-Circuited Evaluations • Formulating Algorithms	Deitel	Chapter 5
03	2/10	Algorithm Development and Control Statements (Part 2) Iteration Statements • Break/Continue • Logical Expressions • Short-Circuited Evaluations	Deitel	Chapter 6
04	2/17	Algorithmic Reviews Practice Problems • Workshops • etc.	Deitel	Chapter 5—6
SAT	2/22	2—3 hours of Coding Test 1 (A) Selections and Iterations	Test 1 (A)	Chapters 3, 5, 6
05	2/24	Methods: A Deeper Look Static Methods • Multiple Arguments • Argument Promotion and Casting • Enumeration • Scope • Activation Records	Deitel	Chapter 7
06	3/2	Arrays Declaring, Creating, Initializing Arrays • Using and Modifying Arrays • Passing Arrays to Methods • Unit-Tests with Collections • Rectangular Arrays • Jagged Array • Handling Exceptions	Deitel	Chapter 8
3/9		Spring Break – March 9-15, 2020	NO CLASS	
07	3/16	Introduction to Classes, Objects, Methods Classes • Objects • Fields • Methods • Properties • Constructors Array and Method Reviews Practice Problems • Workshops • etc.	Deitel	Chapter 4
SAT	3/21	2—3 hours of Coding Test 1 (B) Methods and Arrays	Test 1 (B) Focus	Chapters 6, 7, 8 Arrays
08	3/23	Classes and Objects: A Deeper Look Classes • Objects • Public/Private access modifiers • Constructors • Instantiations • Fields • Methods • Properties • Auto-Implemented Properties Object-Oriented Programming: Inheritance Base classes • Derived classes • Protected Members • Inheritance • Class Object	Deitel Deitel	Chapter 10 Chapter 11
09	3/30	Object-Oriented Programming: Inheritance Constructors in Derived Classes OOP Polymorphism and Interfaces Polymorphism • Examples • Demonstration of Polymorphic Behavior • Abstract Classes and Methods • Sealed Methods and Classes • Creating Using Interfaces	Deitel Deitel	Chapter 11 Chapter 12

Classes		Objectives	Expectations	
10	4/6	ATM Case Study, Part 1: Object-Oriented Design with the UML Requirements Document • Identifying the Classes • Identifying Class Attributes • Identifying Objects' States and Activities • Identifying Class Operations • Identifying Collaboration Among Objects Exception Handling: Deeper Look Divide By Zero without Exception • Handling DlvidebyZeroException and FormatException	Deitel Deitel	Chapter 27 Chapter 13
11	4/13	ATM Case Study, Part 2: Implementing an OO Design in C# Creating Classes • Incorporating Inheritance and Polymorphism • Full OOP Implementations from designs Exception Handling: Deeper Look .NET Exception Hierarchy • finally block • Using statement	Deitel Deitel <i>Test 2 Review</i>	Chapter 28 Chapter 13
SAT	4/18	2—3 hours of Coding Test 2 OOP and Exceptions	Deitel Deitel	Chapters 4, 10—13 Chapters 27 & 28
12	4/20	Using Generics and Collections Motivation for Generic Methods • Collection Classes and Interfaces • Class Array and Enumerators • Dictionary Collections • Null Conditional Operator ?[] • Dictionary and Collection Initializers	Deitel Deitel	Chapter 20.1–2 Chapter 21.1–7
13	4/27	Coding Test 3 Review Practice problems on Collections: Arrays • Lists • Dictionaries		
14	5/4	2—3 hours of Coding Test 3 Methods and Collections (Arrays, Lists, Dictionaries, and Collection Interfaces)	Deitel Deitel	Chapter 21 Chapters 1—8