

|  |
| --- |
| **Software Development I**  **CIS 199-01** |
| **Fall 2016** |

|  |  |
| --- | --- |
| **I. Professor / Instructor** | |
| **Instructor** | Andrew L. Wright, Ph.D. |
| **Contact information** | Office: 304 College of Business  Telephone: 502-852-6098  Fax: 502-852-4799  Email: [andrew.wright@louisville.edu](mailto:andrew.wright@louisville.edu) |
| **Office hours** | Mondays and Wednesdays: 1:30 – 2:30 PM; 4:00 – 5:30 PM Tuesdays and Thursdays: 4:00 – 7:00 PM; and by appointment |

|  |  |
| --- | --- |
| **II. Course Information** | |
| **Class time / Room** | Mondays and Wednesdays 2:30 – 3:45 PM  BS 003 College of Business classroom/DA 304 Davidson Hall computer lab |
| **Required text** | *Starting Out with Visual C#*, 4th Edition, by Tony Gaddis, Pearson/Addison-Wesley, 2017 (ISBN-13: 9780134382609). E-Text is also available. Students are required to bring a USB flash/jump drive to every class. |
| **Course description** | This course introduces object-oriented concepts such as the use of classes, methods, encapsulation, and inheritance. The course concentrates on using object-oriented programming to solve simple problems involving input and output. Computer lab sessions are used to reinforce programming concepts. Extensive programming assignments are required. |
| **Prerequisites** | CIS 100, CIS 150 |
| **Course objectives** | * Introduce the fundamental concepts in computer programming with an emphasis on problem-solving and algorithm design * Introduce the basic features of C#, including control flow, looping, methods, classes, and one-dimensional arrays * Introduce Object-Oriented Programming concepts classes, encapsulation, and inheritance * Explore Graphical User Interface (GUI) programming in C#, including the event model |
| **Learning objectives** | Upon completion of CIS 199, students will be able to:   * explain the fundamental object-oriented (OO) principles of encapsulation and inheritance * write C# programs that demonstrate proper OO design using encapsulation and inheritance * contrast a class with an object * use the Visual Studio Integrated Development Environment to edit, compile, debug, and execute C# applications * contrast syntax errors with logic errors * use the fundamental data types to create variables, including integer, floating point, character, Boolean, and string data * contrast the integral data types with the floating point types * use the basic arithmetic operators to create algebraic expressions * use named constants properly in C# code * contrast the use of value types with reference types * write C# applications that perform input and output using the console * explain the C# GUI event model in simple terms * build GUI-driven C# Windows Forms applications that demonstrate proper GUI design using components including labels, textboxes, and buttons * contrast the use of console applications with those of GUI applications * write simple if, if/else statements * use compound expressions properly in if statements including use of the AND, OR, and NOT operators * write complex nested if/else statements * use the switch statement to make decisions * write simple loops using the while, for, and do/while statements * contrast the use of while, for, and do/while statements * write nested loop statements * contrast definite and indefinite repetition * contrast the use of methods with those of properties * write methods with and without parameters * write methods with and without returning a value * write methods with and without a static modifier * contrast passing simple, value types with passing objects (reference types) to a method * describe the different scopes a variable may have * contrast value, reference, and output parameters * write methods that use value, reference, and output parameters * write applications with overloaded methods and constructors * write methods that use optional parameters * contrast an instance variable with local and static (class) variables * write get/set properties and auto-implemented properties * describe the role of constructors and destructors * implement information hiding through the use of public and private visibility modifiers * declare and initialize a one-dimensional array of simple, value types * declare and initialize a one-dimensional array of objects * write a foreach statement to traverse an array * contrast the use of a traditional loop (such as a for loop) to traverse an array with the use of a foreach loop * declare and initialize a multi-dimensional array * create a new class from an existing class using inheritance * contrast HAS-A and IS-A relationships * contrast public, private, and protected visibility in inheritance * override a method from a base class in a derived class * use base class methods and constructors from a derived class |
| **Final drop date** | October 24, 2016  See: <http://louisville.edu/calendars/academic/undergrad-grad> |
| Expectations of outside time required for class | In addition to direction instruction in the classroom and computer lab, students should expect to spend at least 7.5 hours per week outside of class working on course activities. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **III. Evaluation** | | | | |
| **Grading scale** | A+ = 97 to 100  B+ = 87 to 89.9%  C+ = 77 to 79.9%  D+ = 67 to 69.9% | A = 93 to 96.9%  B = 83 to 86.9%  C = 73 to 76.9%  D = 63 to 66.9%  F = Below 60 | | A- = 90 to 92.9%  B- = 80 to 82.9%  C- = 70 to 72.9%  D- = 60 to 62.9% |
| **Grading scheme** | Grading component | | Weighted grading percentage | |
| 2 Exams | | 30% | |
| Programming assignments and homework | | 30% | |
| Labs, quizzes, and class exercises | | 20% | |
| Comprehensive final exam | | 20% | |
| Total | | 100% | |
|  | In order to pass the class, students must score at least 60% in each category: Exams, Programming Assignments & Homework, Labs, Quizzes & Class Exercises, and Final Exam. For example, a student that earns 55% on exams but 100% on programs and quizzes will still earn an F for the class. Although extra credit may available on some assignments, at no time will it be allowed to elevate any category's contribution beyond the maximum percentages specified above.  Curving may be employed but this minimum will be guaranteed. Note, assignment scores will be posted in Blackboard but Blackboard currently doesn't calculate final weighted scores correctly. Students must use the above category weights to accurately track their progress. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| IV. Schedule | | | |
| Check the Daily Notes for specific plans for each class. Daily Notes are added to Blackboard once or twice a week, in advance of each class. | | | |
| Week | Date | Topic | Readings |
| 1 | 8/22 | Intro to CIS 199 and Tegrity Lecture Capture Tool |  |
| 1 | 8/24 | Intro to Computers and Programming | Chapter 1 |
| 2 | 8/29 | Intro to Computers and Programming; Intro to Visual C#; Lab | Chapters 1-2 |
| 2 | 8/31 | Intro to Visual C# | Chapter 2 |
| 3 | 9/5 | **Labor Day Holiday** |  |
| 3 | 9/7 | Intro to Visual C; Processing Data | Chapters 2-3 |
| 4 | 9/12 | Processing Data; Lab | Chapter 3 |
| 4 | 9/14 | Processing Data | Chapter 3 |
| 5 | 9/19 | Processing Data; Lab | Chapter 3 |
| 5 | 9/21 | Making Decisions | Chapter 4 |
| 6 | 9/26 | Making Decisions; Lab | Chapter 4 |
| 6 | 9/28 | Making Decisions | Chapter 4 |
| 7 | 10/3 | **Fall Break** |  |
| 7 | 10/5 | Making Decisions; Loops, Files, and Random Numbers | Chapters 4-5 |
| 8 | 10/10 | **Exam 1 – Davidson Computer Lab, DA 304** |  |
| 8 | 10/12 | Loops, Files, and Random Numbers | Chapter 5 |
| 9 | 10/17 | Loops, Files, and Random Numbers; Lab | Chapter 5 |
| 9 | 10/19 | Modularizing Your Code with Methods | Chapter 6 |
| 10 | 10/24 | Modularizing Your Code with Methods; Lab | Chapter 6 |
| 10 | 10/26 | Modularizing Your Code with Methods | Chapter 6 |
| 11 | 10/31 | Arrays and Lists; Lab | Chapter 7 |
| 11 | 11/2 | Arrays and Lists | Chapter 7 |
| 12 | 11/7 | Arrays and Lists; Lab | Chapter 7 |
| 12 | 11/9 | More about Processing Data | Chapter 8 |
| 13 | 11/14 | **Exam 2 – Davidson Computer Lab, DA 304** |  |
| 13 | 11/16 | Classes and Multiform Projects | Chapter 9 |
| 14 | 11/21 | Classes and Multiform Projects; Lab | Chapter 9 |
| 14 | 11/23 | **Thanksgiving Holiday** |  |
| 15 | 11/28 | Classes and Multiform Projects; Inheritance and Polymorphism; Lab | Chapters 9-10 |
| 15 | 11/30 | Inheritance and Polymorphism | Chapter 10 |
| 16 | 12/5 | Inheritance and Polymorphism; Review | Chapter 10 |
| 17 | 12/14 | **Final Exam -2:30 – 5:00 PM CIS Computer Lab, BS 054 (note room!)**  **Also see:** [**http://louisville.edu/registrar/registration-information/final-exam-schedules**](http://louisville.edu/registrar/registration-information/final-exam-schedules) |  |

|  |  |
| --- | --- |
| **V. Additional Work Details** | |
| **Programming assignments** | Number: A maximum of 7 programming assignments will be given. Programs will be assigned and discussed during class, so attendance is important.  Grading: The score for each program will depend on  1. Correctness of program logic and output (~ 60%)  2. Program style - variable names, indenting, output formats, etc. (~ 20%)  3. Documentation - see below (~ 20%)  Documentation: Each source file (edited by the student) should contain  1. Your **grading ID** (not your name nor your student ID), assignment due date, course number and section, and program number  2. A brief description of what the program does  3. A brief description and a variable dictionary for every method and class written, including preconditions and postconditions for methods, properties, and constructors  4. Comments which will identify and explain important sections in the code  5. Magic Numbers (decimal literals) are to be avoided. Use named constants instead.  Any missed documentation will result in a 1 point deduction per occurrence. For example, if you forgot to describe 5 variables in your program, you would immediately lose 5 points regardless of whether your program works or not. Words to live by: "When in doubt, document it!!!"  Submission: Programs should be handed in at the time specified by the assignment and must contain  1. All source code files needed for compilation (named as specified) will be submitted using Blackboard's Assignment tool. Files may be e-mailed to your instructor only if problems prevent the use of Blackboard. Be sure to keep a backup of all your files!!!  2. Printouts of all source code created or modified for the assignment, if requested by the assignment. Electronic submission and grading will be used unless otherwise specified.  Lateness: No late programs will be accepted unless agreed upon in advance by the instructor. |
| **Interviews** | The instructor reserves the right to require an interview with the student before assigning a grade for a programming assignment. If the student is unable to adequately explain how the program works, the student will receive a **zero** for the assignment regardless of how well the program runs. Students may be selected at random or upon suspicion of cheating. Once notified, students have up to one week to schedule an interview. |
| **Quizzes** | Quizzes and in-class exercises will not generally be announced in advance. Missed in-class work may not be made up without convincing reasons (such as a doctor's note) for the absence.  All quizzes are closed book and closed notes unless otherwise specified. |
| **Exams** | Missed exams count as zero (0) points. A make-up test is allowed only if convincing reasons (such as a doctor's note) are given for the absence. If possible, arrange **in advance** of test.  All exams may include an in-class portion and a take-home portion at the instructor's discretion.  Students will be allowed to bring one 8.5" x 11" page (front only) "cheat sheet" to the exams. You may include anything you want (such as sample programs, worked homework problems, definitions, etc.) that will fit on the cheat sheet. Students may bring one 8.5" x 11" page, front and back, to the final exam.  Test dates are given in the TENTATIVE Schedule. Any change will be announced as far in advance as possible (usually at least one week). Any such changes to the course schedule will be published to the *Syllabus* section of the course in Blackboard and explained in a Blackboard announcement. |
| **Participation and class contribution** | Attendance of every class is encouraged. From time to time, missing a class is inevitable. If this happens, the student is responsible for what was covered in class and obtaining class handouts. Students are expected to arrive to class on time. The instructor will also strive to start class on time. If, however, the instructor is late by more than 15 minutes, the class will stand as cancelled. |

|  |  |
| --- | --- |
| **VI. Student Responsibilities / College and University Issues** | |
| **University of Louisville student**  **conduct and responsibilities** | This course will abide by University of Louisville student  conduct and responsibilities with regards to ethics and related issues:  <http://louisville.edu/dos/students/policies-procedures/student-handbook.html#codeofstudentconduct> |
| **College of Business student conduct and responsibilities** | This course will abide by College of Business student  conduct and responsibilities with regards to ethics and related issues:  <http://business.louisville.edu/students/college-of-business-academic-dishonesty-policy>  Programming intensive classes present unique problems. Do your own work. While some discussion of algorithms between students may be helpful and acceptable, do not copy another person's program. This is cheating and will be dealt with severely. |
| **Religious holiday conflict policy** | <http://louisville.edu/diversity/diversity-resources-for-students/work-restricted-holiday-calendar/view> |
| **University policy on equal access** | <https://louisville.edu/disability/students/ada-policy-formal-complaint-procedure-legislation> |
| **Diversity** | The University of Louisville strives to foster and sustain an environment of inclusiveness that empowers us all to achieve our highest potential without fear of prejudice or bias. We commit ourselves to building an exemplary educational community that offers a nurturing and challenging intellectual climate, a respect for the spectrum of human diversity, and a genuine understanding of the many differences-including race, ethnicity, gender, gender identity/expression, sexual orientation, age, socioeconomic status, disability, religion, national origin or military status-that enrich a vibrant metropolitan research university. We expect every member of our academic family to embrace the underlying values of this vision and to demonstrate a strong commitment to attracting, retaining and supporting students, faculty and staff who reflect the diversity of our larger society. For more information, visit the [Office of Diversity](http://louisville.edu/diversity/). |
| **Title IX/Clery Act Notification** | Sexual misconduct (including sexual harassment, sexual assault, and any other nonconsensual behavior of a sexual nature) and sex discrimination violate University policies. Students experiencing such behavior may obtain **confidential** support from the PEACC Program (852-2663), Counseling Center (852-6585), and Campus Health Services (852-6479). To report sexual misconduct or sex discrimination, contact the Dean of Students (852-5787) or University of Louisville Police (852-6111).  Disclosure to **University faculty or instructors** of sexual misconduct, domestic violence, dating violence, or sex discrimination occurring on campus, in a University-sponsored program, or involving a campus visitor or University student or employee (whether current or former) is **not confidential** under Title IX. Faculty and instructors must forward such reports, including names and circumstances, to the University’s Title IX officer.  For more information, see the Sexual Misconduct Resource Guide  ( <http://louisville.edu/hr/employeerelations/sexual-misconduct-brochure> ). |
| **Classroom policies** | Please ensure that all cellular phones, pagers, and other electronic devices are turned off or placed in vibrate mode **before** entering class. Disruptions of class will be considered inappropriate conduct.  Exceptions will not be granted nor grades of "Incomplete" given except in accordance with applicable University and College of Business policies.  Along with preparing for and attending class, each student has the responsibility to promote high academic standards. Students are expected to cooperate in all classes with faculty members to achieve an optimal learning environment. Inappropriate classroom behavior may result in the student being withdrawn from the course, and potentially assigned academic penalties. Inappropriate classroom behavior will be dealt with in the same manner as academic dishonesty. |
| **Lecture capture** | Your instructor uses a combination of software, microphones and/or cameras to create a recording of most class sessions. These recordings are generally only made available to students enrolled in this class and authorized individuals, such as teaching assistants. In rare circumstances, such as when a recording cannot be created in one section, a recording may be made available to another section of the same course. Be aware that student voices may be captured as part of any recordings made in the classroom. |
| **Changes in the syllabus** | The schedule and procedures for grading in this course are subject to change in the event of extenuating circumstances, such as severe weather or pandemic illness. Any material changes to the course syllabus will be published to the Syllabus section of the course in Blackboard and explained in a Blackboard announcement. |

Updated: 8/23/2016