**a**



|  |
| --- |
| **Software Development II**  **CIS 200-01** |
| **Spring 2017** |

|  |  |
| --- | --- |
| **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: 2:30 – 4:00 PM; Tuesdays and Thursdays: 2:30 – 4:30 PM; and by appointment |

|  |  |
| --- | --- |
| **II. Course Information** | |
| **Class time / Room** | Tuesdays and Thursdays 1:00 – 2:30 PM  BS 003 College of Business classroom/DA 304 Davidson Hall computer lab |
| **Required text** | *Visual C#: How to Program*, 6th Edition, by Paul Deitel & Harvey Deitel, Prentice Hall, 2017 (ISBN-13: 9780134601540). E-Text is also available. Students are required to bring a USB flash/jump drive to every class. |
| **Course description** | This course emphasizes object-oriented software development. Students study the object model and apply it to systems development problems. Topics include polymorphism, inheritance, and object interaction. Event-driven programming of graphical user interfaces is introduced. Application areas may include data structures, searching, sorting, and databases. Extensive programming assignments are required. |
| **Prerequisites** | CIS 199 |
| **Course objectives** | * Introduce Object-Oriented Programming concepts encapsulation, inheritance, and polymorphism * Explore Graphical User Interface (GUI) programming in C#, including the event model * Introduce exception handling in C# * Explore basic file processing techniques in C# * Introduce simple data structures and collections, including stacks, queues, linked lists, and trees * Explore recursion * Explore algorithm efficiency * Study sorting algorithms including selection sort, insertion sort, and merge sort * Introduce generic methods and generic methods * Introduce databases and LINQ |
| **Learning objectives** | Upon completion of CIS 200, students will be able to:   * explain the fundamental object-oriented (OO) principles of encapsulation, inheritance, and polymorphism * write C# programs that demonstrate proper OO design using encapsulation, inheritance, and polymorphism * contrast the use of interfaces with abstract classes * write simple LINQ statements in C# to query program objects * contrast declarative programming (such as used in LINQ) with imperative programming * explain how exception handling works in C# * use exception handling in C# programs to build robust, fault-tolerant applications * explain the C# GUI event model and contrast it with prior models used * build GUI-driven C# Windows Forms applications that demonstrate proper GUI design using components including labels, textboxes, buttons, menus, dialog boxes, and combo boxes * classify the levels in the hierarchy of data * contrast the use of text files with the use of object serialization * build C# applications that read and write data using simple text files * build C# applications that read and write data using object serialization * classify algorithms based on their algorithmic efficiency (Big O) * contrast the linear search algorithm with the binary search algorithm * contrast the use of selection, insertion, merge, and quick sort algorithms * choose an appropriate sorting algorithm to use for specified conditions * contrast static data structures with dynamic data structures * explain the operation of self-referential structures, including linked lists * contrast the stack and queue data structures * explain the operation of binary tree data structures, including the binary search tree * write C# applications that extend the functionality of the fundamental data structures * explain the operation of the built-in collections available in C#, including the hash table * explain the advantages of writing generic methods and classes * write simple generic methods and classes in C# including constraints * explain the operation of recursive algorithms * analyze the algorithmic efficiency of simple recursive methods (primarily using tail recursion) * contrast the use of recursive methods and non-recursive methods for solving the same problem * write simple methods in C# that use recursion properly * learn about the relational database model * use an ADO.NET Entity Data Model to create classes for interacting with a database via LINQ to Entities * use LINQ to retrieve and manipulate data from a database * add data sources to projects * use the IDE’s drag-and-drop capabilities to display database tables in apps * use data binding to move data seamlessly between GUI controls and databases * create Master/Detail views that enable you to select a record and display its details |
| **Final drop date** | March 9, 2017  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 | | 15% | |
| Comprehensive final exam | | 25% | |
| 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 | 1/10 | Introduction and Review |  |
| 1 | 1/12 | Classes and Objects: A Deeper Look | Chapter 10 |
| 2 | 1/17 | Classes and Objects: A Deeper Look; OOP: Inheritance | Chapters 10-11 |
| 2 | 1/19 | OOP: Inheritance | Chapter 11 |
| 3 | 1/24 | OOP: Polymorphism | Chapter 12 |
| 3 | 1/26 | OOP: Polymorphism | Chapter 12 |
| 4 | 1/31 | Intro to LINQ and the List Collection | Chapter 9 |
| 4 | 2/2 | Intro to LINQ and the List Collection; Exception Handling: A Deeper Look | Chapters 9, 13 |
| 5 | 2/7 | Exception Handling: A Deeper Look; GUI with Windows Forms | Chapters 13-14 |
| 5 | 2/9 | GUI with Windows Forms | Chapter 14 |
| 6 | 2/14 | **Exam 1 – Davidson Computer Lab, DA 304** |  |
| 6 | 2/16 | GUI with Windows Forms | Chapters 14-15 |
| 7 | 2/21 | GUI with Windows Forms | Chapter 15 |
| 7 | 2/23 | GUI with Windows Forms; Files and Streams | Chapters 15,17 |
| 8 | 2/28 | Files and Streams | Chapter 17 |
| 8 | 3/2 | Recursion | Section 7.15 |
| 9 | 3/7 | Searching and Sorting | Chapter 18 |
| 9 | 3/9 | Searching and Sorting | Chapter 18 |
| 10 | 3/14 | **Spring Break** |  |
| 10 | 3/16 | **Spring Break** |  |
| 11 | 3/21 | Searching and Sorting | Chapter 18 |
| 11 | 3/23 | Data Structures | Chapter 19 |
| 12 | 3/28 | **Exam 2 – Davidson Computer Lab, DA 304** |  |
| 12 | 3/30 | Data Structures | Chapter 19 |
| 13 | 4/4 | Data Structures | Chapter 19 |
| 13 | 4/6 | Data Structures; Generics | Chapters 19-20 |
| 14 | 4/11 | Generics; Collections | Chapters 20-21 |
| 14 | 4/13 | Collections; Databases and LINQ | Chapters 21-22 |
| 15 | 4/18 | Databases and LINQ | Chapter 22 |
| 15 | 4/20 | Databases and LINQ and Review | Chapter 22 |
| 17 | 5/2 | **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. The grading ID is a randomly generated identifier that allows for anonymous grading. It will appear in Blackboard under My Grades after drop/add is complete.  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 must be submitted at the time specified by the assignment and must contain all source code files needed for compilation (named as specified), usually in the form of a single ZIP file containing the project’s files and folders. Files will be submitted in Blackboard. Files may be e-mailed to your instructor only if problems prevent the use of Blackboard for submission. Be sure to keep a backup of all your files!!!  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 will be a mix of those completed online, in Blackboard, and those done in-class. Pop 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 in-class quizzes and exercises 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: 1/7/2017