REGIS UNIVERSITY

COLLEGE OF COMPUTER & INFORMATION SCIENCES

Syllabus

Course Number: CS 362
Course Title: Data Structures

Course Description

CS 362 – DATA STRUCTURES (3). Combines concepts discussed in CS361, computer programming and design, with structural programming and design. Examines data structures including arrays, structures, linked lists, queues, stacks, file organization along with file processing and algorithms used in problem solving. Emphasizes sound programming practices.

Prerequisite Courses

CS361 Control Structures

In order to successfully complete this course, students are expected to have taken the course prerequisite, CS361 (or transferred in an equivalent). From CS361, you should have a working knowledge of the following topics:

1. Practice with problem definition, solution construction and algorithmic development.
2. Practice with top-down design techniques, coding, and debugging.
3. Ability to implement code with C++ control structures used for decisions and iteration, including: if, while, do-while, for, and switch.
4. Understanding of modular code design with C++ functions, and proper parameter passing, using both pass-by-value and pass-by-reference.
5. Understanding of recursion.

Course Overview

In the pre-requisite programming course (CS361), students learned how to use various control structures (if-then, while-do, functions, etc) to control the flow of the program.

In this course, students will learn how to implement various data structures (enumerated types, arrays, records/structs, and linked lists) and how to use files (text and binary) to store and manipulate program data. Good programming practices will be encouraged throughout this course.

Ultimately, programming is a skill that requires the ability to put designs into practice. The only way to succeed is to practice this skill. Therefore, this course will require a ***significant*** amount of time each week to complete the programming assignments. The workload for the first couple of weeks will be ***at least*** as much as the workload in the last couple of weeks in CS361, and will

increase further in the last half of the course. If you are not willing or able to spend the necessary time, please reconsider whether this is the correct time to attend this class.

Course Outcomes

Upon completion of this course, students should be able to:

1. Compare and contrast the data representation employed by text files and by binary files, and implement the usage of both for program data storage.
2. Define and implement Enumerated data types and Record data structures within high-level language programs.
3. Compare and contrast single and multi-dimensional arrays, and implement both within high-level language programs.
4. Compare and contrast procedural programming concepts with object-oriented programming concepts.
5. Define and use simple Classes and Objects within high-level language programs.
6. Create linked lists implemented using dynamically allocated memory within high-level language programs.
7. Differentiate between Stack and Queue data structures, and compare alternative implementations with respect to performance.
8. Choose the appropriate file and data structures for modeling a given programming problem.
9. Design, implement, test, and debug modular high-level language programs, which implement the file and data structures introduced in this course, to logically solve typical programming tasks.

Course Materials

Required Texts:

Malik, D.S. (2015). *C++ Programming: From Problem Analysis to Program Design* (7th ed.). Boston, MA: Cengage Learning; ISBN-13: 978-1285852744, ISBN-10: 1285852745

Textbook Notes:

- This is the same textbook that was used in CS361. CS362 will cover the second half.

Required Technology Tools:

1. A PC-compatible computer system running Windows.

2. A C++ Compiler -- the preferred compiler is the free **Dev-C++** compiler

   The newest version of Dev-C++ (Orwell), can be downloaded from:
   http://sourceforge.net/projects/orwelldevcpp/files/Setup%20Releases/

Download the most recent version:
Dev-Cpp x.x.x MinGW x.x.x Setup.exe (32-bit version)
OR
Dev-Cpp x.x.x TDM-GCC x64 x.x.x Setup.exe (64-bit version)

WARNING: If you had an older version loaded on your computer from an earlier class, such as the Bloodshed Dev-C++ compiler, ***you must uninstall it*** before installing the newer version.

Other ANSI standard C++ compilers can be used, but the facilitator will be most familiar with the Dev-C++ compiler listed above, and will most likely use the Dev-C++ compiler to grade your programs. If a student chooses to use another compiler, it is the student's responsibility to obtain any help needed to use the software, and to insure that all assigned programs compile and run correctly on the Dev-C++ compiler.

As with most of Regis learning activities, using various software applications to accomplish assignments requires students to exercise a great deal of responsibility for learning how to successfully operate the software applications. This class assumes that you have a working knowledge of the C++ compiler from CS361. There are, however, many quality books on the market that support novice users if you need them.

Pre-Assignment

Complete the following tasks:

- Read the first week's assigned reading in the textbook (listed in the Course Assignments and Activities grid on the next page).
- Be prepared to **ask questions** on unclear areas and **respond** to questions about information in the assigned reading.

Online Format: Sign on to worldclass.regis.edu and become familiar with the course navigation of the Web Curriculum. Complete assignments above.

Classroom-based Format: Complete assignments above by the first night of class.

Course Assignments and Activities

	Topics	Readings (7^{th} edition of textbook)*	Activities Assignments and Associated Points**
1	Introduction to Data Structures Text Files Single Dimensional Arrays	**Various chapters:** pp. 53 – 54, 156 – 157, 203 – 204 (string review) **Chap 3**, pp. 160 – 174 (file I/O and review) **Chap 4**, pp. 217 – 219 (read input failure) **Chap 6**, pp. 423 – 432 (program example - file I/O) **Chap 8**, pp. 520 – 542 (one-dimensional arrays), pp. 576 – 588 (program examples), and pp. 557 – top of 558 (specifying filenames at execution) ***Optional:*** Odd exercises 19 – 25 on pages 179 – 180, and odd exercises 1 (a-i) – 29 on pages 591 – 598 (*answers in back of text*).	Participation in Discussions 10% for entire course Programming Assn 1 (file input and arrays) 10% of 52% NOTE: Together, all 6 programming assignments will be worth 52% of the course grade.
2	Enumerated Data Types Sort/Search using Arrays Structure Charts	**Chap 7**, pp. 466 – top of 485 (enumerated types/typedef and sample program) **Chap 8**, pp. bottom of 542 – middle of 549 (sequential search/selection sort), middle of 558 – middle of 559 (parallel arrays) **Chap 16**, pp. 1056 – middle of 1057 (sequential search), 1065 – 1068 (binary search), and 1077 – 1091 (parallel arrays program example) **Appendix F**, pp. 1328 – 1329 (character checking library review) ***Optional:*** Exercises 1 (a - h), 3, and 5 on pages 510 – 511, and exercises 1, 3, and 15 on pages 1093 – 1095 (*answers in back of text*)	Participation in Discussions
3	Records Library Functions	**Chap 7**, pp. bottom of 490 – 510 (string functions, example, and review) **Chap 9**, all (records/structs) ***Optional:*** Exercises 15 and 17 on pages 514 – 515, and odd exercises 1 - 17 on pages 641 – 645 (*answers in back of text*).	Participation in Discussions Programming Assn 2 (file I/O, enum types, parallel arrays, search/sort) 16% of 52%

4	Multi-Dimensional Arrays	**Chap 8**, pp. middle of 559 – 590 (two- and multi-dimensional arrays and review) ***Optional:*** Odd exercises 39 - 43 on pages 601 – 602 (*answers in back of text*).	Participation in Discussions Programming Assn 3 (array of records) 20% of 52% Midterm Exam 19 %
5	Pointers Structured Binary Files Dynamic Memory Allocation	**Chap 12**, pp. 812 – 841 (pointers and dynamically allocated memory) **Appendix E**, pp. middle of 1310 – 1323 (binary files)	Participation in Discussions Programming Assn 4 (2D arrays and binary files) 16% of 52%
6	Data Abstraction Classes and Objects	**Chap 10**, 650 – 698, 705 – 720 (classes, objects, data abstraction, program examples and review) **Chap 12**, pp. bottom of 841 – 852 (classes and pointers: peculiarities) **Chap 13**, 888 – middle of 952 (operator overloading and program example) ***Optional:*** Odd exercises 1 – 17 on pages 720 – 726, odd exercises 1 - 33 on pages 873 – 880, and odd exercises 1 – 11, 21 – 25 on pages 965 – 969 (*answers in back of text*).	Participation in Discussions Programming Assn 5 (classes/objects) 16% of 52%
7	Classes and Objects (templates) Linked Lists	**Chap 13,** 953 – 965 (templates and review) **Chap 17**, pp. 1058 – 1103 (linked lists) ***Optional:*** Odd exercises 29 – 37 on pages 969 – 971, and odd exercises 1 - 19 on pages 1182 - 1187 (*answers in back of text*).	Participation in Discussions
8	Stacks & Queues	**Chap 18**, pp. 1196 – 1229 (stacks) and middle of 1245 - 1262 (queues) **Quick Reviews** for chapters 3, 7, 8, 9, 10, 12 (1-27), 13, 17 (1-9) and 18 (1-6, 10-15) ***Optional:*** Odd exercises 1 – 7 on page 1282 – 1284 (*answers in back of text*).	Participation in Discussions Programming Assn 6 (linked lists) 22% of 52% Final Exam 19 %
		Total	**100%**

****Other Edition Reading Assignments:*** Alternate reading assignments for the previous edition of the textbook (6th edition) will be available in your facilitator's syllabus.

*****Note to Classroom sections only:*** Exact dates for reading assignments and programming assignments may be different than indicated in the above Course Assignments and Activities grid. Your facilitator's syllabus, handed out the first night of class, will indicate any changes.

Summary of Assignments and Percentage Weight towards course grade

Assignment		Value (percent of overall course grade)
Programming Assignments (6)		
Program 1 (file I/O and arrays)	10% of 52%	
Program 2 (enum types, parallel arrays, sorting and searching)	16% of 52%	
Program 3 (array of records)	20% of 52%	
Program 4 (2D arrays and binary files)	16% of 52%	
Program 5 (classes and objects)	16% of 52%	
Program 6 (linked lists)	22% of 52%	
Total for all Programming Assignments		52%
Midterm Exam		19 %
Final Exam		19 %
Forum / Participation		10 %
Course Total		100 %

Course Policies and Procedures

Programming Assignments
Each programming assignment will involve writing a program that implements the concepts discussed in the book and class. The percentage of the grade allocated to each programming assignment is an indication of the relative effort that will be required for implementation.

> ***Late Assignment Policy for CS362 Programming Assignments***
> Late programming assignments will be graded and then 2% will be deducted for each day the assignment is late, ***up to 5 days late***. No programming assignment will be accepted more than ***5 days*** after the official due date. Therefore, any programming assignment turned in more than ***5 days*** late will be given a grade of **zero**, and no feedback will be given.

Exams
There will be a midterm exam and a final exam. Exam questions will be cumulative, taken from reading assignments and course content. Exams will **not** be accepted late.

Adding this course during the Drop/Add Period

If you added this course during the drop/add period, after class began on Monday, you are responsible for ***immediately*** notifying the instructor that you joined the course late. None of the course due dates will be extended for you. Even if a due date already passed when you added the course, late points will still be deducted.

Repeating the course

If you are repeating this course (due to a previous withdraw or low grade), you are responsible for ***immediately*** notifying the instructor. Course assignments that you submitted when you last took the course cannot be repeated -- you will be required to complete alternate assignments.

Plagiarism

Plagiarism includes submitting code or anything else that was obtained from any other person, publication, or any internet web source. ***All work submitted in CS362 must be your own.***

In cases of suspected cheating or plagiarism, the instructor will discuss the matter with the student(s) involved. The instructor reserves the right to question any student orally or in writing about any assignment, and to use the evaluation of the student's understanding of the assignment and of the submitted solution as evidence of cheating. All cheating incidents will be reported to the Computer Science department, and may also be reported to the Academic Integrity Board for further action.

CC&IS Grading Scale

Letter Grade	Percentage	Grade Point
A	93 to 100	4.00
A–	90 to less than 93	3.67
B+	87 to less than 90	3.33
B	83 to less than 87	3.00
B–	80 to less than 83	2.67
C+	77 to less than 80	2.33
C	73 to less than 77	2.00
C–	70 to less than 73	1.67
D+	67 to less than 70	1.33
D	63 to less than 67	1.00
D-	60 to less than 63	.67
F	Less than 60	0

Additional information about grading can be found in the latest edition of the University Catalog, available at http://www.regis.edu/Academics/Course%20Catalog.aspx.

CC&IS Policies and Procedures

Each of the following CC&IS Policies & Procedures is incorporated here by reference. Students are expected to review this information each term, and agree to the policies and procedures as identified here and specified in the latest edition of the University Catalog, available at http://www.regis.edu/Academics/Course%20Catalog.aspx or at the link provided.

- The CC&IS Academic Integrity Policy.
- The Student Honor Code and Student Standards of Conduct.
- Incomplete Grade Policy, Pass / No Pass Grades, Grade Reports.
- The Information Privacy policy and FERPA. For more information regarding FERPA, visit the U.S. Department of Education.

- The HIPPA policies for protected health information. The complete Regis University HIPAA Privacy & Security policy can be found here: http://www.regis.edu/About-Regis-University/University-Offices-and-Services/Auxiliary-Business/HIPAA.aspx.
- The Human Subjects Institutional Review Board (IRB) procedures. More information about the IRB and its processes can be found here: http://regis.edu/Academics/Academic-Grants/Proposals/Regis-Information/IRB.aspx.

The CC&IS Policies & Procedures Syllabus Addendum summarizes additional important policies including, Diversity, Equal Access, Disability Services, and Attendance & Participation that apply to every course offered by the College of Computer & Information Sciences at Regis University.

A copy of the CC&IS Policies & Procedures Syllabus Addendum can be found here: https://in2.regis.edu/sites/ccis/policies/Repository/CCIS%20Syllabus%20Addendum.docx.