

Syllabus

Course Number: CS 372

Course Title: Advanced Programming and Algorithms

Course Description

CS372 – ADVANCED PROGRAMMING AND ALGORITHMS (3). Develops advanced programming skills. Presents advanced features of program design and analysis, group programming, and large projects design and implementation. Introduces analysis of computer algorithms from an efficiency point of view.

Prerequisite Courses

**CS208 Computer Science Fundamentals
and
CS362 Data Structures**

In order to successfully participate in this course, students are expected to have completed the course prerequisites. The prerequisites for CS372 are CS208 and CS362, and the prerequisite for CS362 is CS361. Therefore, students **must have completed CS208, CS361 and CS362** (or their equivalents) and have a working knowledge of the following topics within each course:

1. Practice with problem definition, solution construction and algorithmic development using top-down design techniques (CS208/CS361/CS362).
2. Ability to implement code with C++ control structures used for decisions and iteration, including: if, while, do-while, for, and switch (CS361).
3. Understanding of modular code design with C++ functions, and proper parameter passing, using both pass-by-value and pass-by-reference (CS361/CS362).
4. Practice with using files to store and access data (CS362).
5. Ability to implement code with C++ data structures, including: enumerated types, arrays, structures/records, and linked lists, and the ability to choose appropriate data structures to store data for a particular programming problem (CS362).

Course Overview

The previous two programming courses (CS361 and CS362) showed how to use various control structures (if-then, while-do, functions, etc) to control the flow of the program and how to use various data structures (enumerated types, arrays, records, linked lists, etc) to store program data.

This course will explore several additional aspects of programming. The new concepts to be covered in CS372 are:

- Further expansion of your programming skills
- An introduction to Algorithm Analysis
- Searching and Sorting methods
- An introduction to Binary Trees
- Team Program Development and Implementation

CS372 is challenging and covers a substantial amount of material at a rapid pace. The only way to succeed in programming is to practice. Programming can require a *significant* amount of time each week. There will also be **team projects** in this course, which will demand timeliness. If you are not willing or able to spend the necessary time for programming and team coordination, please reconsider whether this is the correct time to attend this class.

Course Outcomes

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

1. Describe operations on and common applications for linked lists, hash tables, binary search trees, stacks, and queues, and implement them in a high-level programming language.
2. Explain the use of big-O notation to describe the amount of work done by an algorithm and to determine the bounds on time complexity.
3. Compare and contrast various searching methods and their computational efficiencies, and design and implement a hashing function and hash table collision-resolution algorithm appropriate to a specific search application.
4. Describe and implement the most common *quadratic* and $O(N \log N)$ sorting algorithms, including selection, insertion, bubble, shell, merge and quick sorts, and compare their computational efficiencies and relative advantages.
5. Discuss factors other than computational efficiency that influence the choice of algorithms, and be able to evaluate, select and justify the selection of appropriate algorithms for a common programming problem.
6. Describe the divide-and-conquer approach, and implement, test and debug recursive algorithms to solve programming problems.
7. As a team member, work effectively to design and implement a programming solution, and demonstrate the central elements of team building and team management.

Course Materials

Required Texts:

Malik, D.S. (2010). *Data Structures using C++* (2nd edition). Boston, MA: Course Technology; ISBN-13: 978-0-324-78201-1.

Technology Tools:

1. A PC-compatible computer system running an MS Windows operating system.
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. For team projects, the Dev C++ compiler will be the default, unless **all** team members agree on using another 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 a working knowledge of the C++ compiler from CS362. There are, however, many quality books on the market that support novice users if a student needs them.

Pre-Assignment:

Students will be **prepared to ask questions** on unclear areas and to **respond** to questions about information in the assigned reading for week 1 (listed in grid on next page).

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:

Wk	Topics	Readings*	Activities Assignments and Associated Points**
1	<ul style="list-style-type: none"> • C++ Concepts • ADTs and Linked Lists • Stacks and Queues 	Chap 1: pgs 1 – middle of 8 (SW development); Chap 3: pgs 132 – top of 155 (pointer review); Chap 5: pgs 266 – 310 (linked list review); Chap 7, all (stacks); Chap 8, all (queues) Optional: Chap 1: 18-51 (classes)	Participation in Discussions (10% for all 8 weeks) Programming Assn 1 (individual – stacks and queues) 10%
2	<ul style="list-style-type: none"> • ADTs and Linked Lists (variations) • Algorithm Efficiency and Big-O Notation 	Chap 5: pgs 310 – 320 (doubly linked lists) Chap 1: pgs 8 – 17 (big-O)	Participation in Discussions Programming Assn 2 (individual - doubly linked lists) 10%
3	<ul style="list-style-type: none"> • Large Projects • Basic Searching Algorithms • Searches with Hashing 	Chap 9: all (searching/hasing)	Participation in Discussions Programming Assn 3 Design (team – searching/hasing)
4	<ul style="list-style-type: none"> • Searches with Hashing (analysis) • Recursive Algorithms 	Chap 6: all (recursion)	Participation in Discussions Programming Assn 3 (team – searching /hasing) 10% Midterm Exam 20%
5	<ul style="list-style-type: none"> • Simple Sorting Algorithms • C++ Concepts (ptrs to functions) 	Chap 10: pgs 534 – top of 552 (comparison sorts) Appendix F: pgs 825 – 826 (insertion sort analysis)	Participation in Discussions Programming Assn 4 Design (team – sorting)
6	<ul style="list-style-type: none"> • Complex Sorting Algorithms 	Chap 10: 552 – 567 (complex sorts) Appendix F: pgs 826 – 832 (quick sort analysis)	Participation in Discussions Programming Assn 4 (team – sorting) 10%
7	<ul style="list-style-type: none"> • Binary Search Trees 	Chap 11: pgs 600 - 608 (binary trees), 616 – top of 628 (binary search trees)	Participation in Discussions
8	<ul style="list-style-type: none"> • Advanced Algorithm Issues 		Participation in Discussions Programming Assn 5 (individual – binary trees) 10% Final Exam – 20%
Total			100%

***Alternate Reading Assignments:** Alternate reading assignments for the previously required textbooks will be available in your facilitator's syllabus, in case you purchased them previously.

****Note to Classroom sections only:** Exact due dates for programming assignments may be modified from what is 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)
<i>Programming Assignments (5)</i>	
Program 1 (individual)	10 %
Program 2 (team)	10 %
Program 3 (individual)	10%
Program 4 (team)	10%
Program 5 (individual)	<u>10 %</u>
<i>Programming Total</i>	50 %
Midterm Exam	20 %
Final Exam	20 %
Participation in Discussions (Classroom: and other class activities)	10 %
Totals	100 %

Course Policies and Procedures

Programming Assignments

Each programming assignment will involve designing and writing a program, using the concepts discussed in the book and class, and may also include submission of analyses, teammate evaluations, structure charts, test data and/or test plans. **Note:** Programs that do not compile, are not modular, or that use any global variables, will NOT be accepted.

One of the goals of this class is to learn how to design and implement a program as part of a programming team. Therefore, some of your assignments will be team-programming assignments. Note that although students will work in teams, the grades will be individual.

CS372 Late policies

Due to the requirement to forms teams in this class, there will be a strict adherence to the late policy. Programming assignments will be graded, and then 2% will be deducted *for each day* the assignment is late, up to the maximum allowed. Programming assignments turned in more than 1 day late (for assns 1-3) or 4 days late (for assns 4-5) 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 individual assignments submitted in CS372 must be your own. And team assignments must be the work of the team member only.***

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](http://www.ed.gov).
- The HIPAA 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>.