

Syllabus

Course Name	Data Structures & Algorithms	Current Semester	Spring 2012-13																								
Course Number	CSIS 237	Credits	3																								
Section Number	01 (eSIS Class# 5891)	Class Time & Place	11:00-11:50 MWF, SH 12 A																								
Instructor	Dr. Anthony Varghese	Office	South Hall 127C																								
Office Tel.	715-425-3335	Fax Number	715-425-0707																								
Email	anthony.varghese@uwrf.edu	Office Hours	MWF 8-10am, MW 1-2pm.																								
Prerequisites:	Knowledge of object-oriented programming using the Java language. CSIS 235 or equivalent is required.																										
Prior Technology Expectations:	An operating system - Windows, Linux, or Mac OS X - and a web browser. Eclipse IDE and Java JRE version 6-8. A word processor: OpenOffice/Microsoft Word.																										
Textbook	<u>Absolute C++ 4th Ed</u> , by Walter Savitch																										
Course Website	Web access will be through the Desire2Learn website: http://www.uwrf.edu/desire2learn/ . Use the D2L site to see the latest announcements, course information, and assignments, and the instructor's feedback on the labs and assignments you submit.																										
Course Description:	This course is an introduction to data structures using the C++ language. We will design and implement data structures such as Linked Lists, Stacks, Queues, Trees, Graphs, and Hash Tables. We will analyze the performance of algorithms and study algorithm design.																										
Course Objective:	By the end of this course, you will be able to design, implement and analyze abstract data types like linked-lists, stacks, queues, and binary trees using an object-oriented approach.																										
Grading Criteria:	<table><tr><td>4 Programming Assignments</td><td>30%</td><td>300 points</td></tr><tr><td>Best 12 of 13 Quizzes</td><td>30%</td><td>300 points</td></tr><tr><td>Best 10 of 11 Online quizzes</td><td>15%</td><td>150 points</td></tr><tr><td>Best 9 of 10 Labs</td><td>22.5%</td><td>225 points</td></tr><tr><td>Final</td><td>2.5%</td><td>25 points</td></tr></table>			4 Programming Assignments	30%	300 points	Best 12 of 13 Quizzes	30%	300 points	Best 10 of 11 Online quizzes	15%	150 points	Best 9 of 10 Labs	22.5%	225 points	Final	2.5%	25 points									
4 Programming Assignments	30%	300 points																									
Best 12 of 13 Quizzes	30%	300 points																									
Best 10 of 11 Online quizzes	15%	150 points																									
Best 9 of 10 Labs	22.5%	225 points																									
Final	2.5%	25 points																									
Evaluation Methods:	<p>Quizzes in class are closed-book.</p> <p>Final letter grades will be based on total percentage scores:</p> <table><tr><td>>= 94%</td><td>A</td><td>72.5-77.4%</td><td>C</td></tr><tr><td>90.0-93.9%</td><td>A-</td><td>70.0-72.4%</td><td>C-</td></tr><tr><td>87.5-89.9%</td><td>B+</td><td>67.5-69.9%</td><td>D+</td></tr><tr><td>82.5-87.4%</td><td>B</td><td>60.0-67.4%</td><td>D</td></tr><tr><td>80.0-82.4%</td><td>B-</td><td><60%</td><td>F</td></tr><tr><td>77.5-79.9%</td><td>C+</td><td></td><td></td></tr></table> <p>If you have a question about an assignment or exam grade, please see me in person within two weeks after receiving graded work.</p>			>= 94%	A	72.5-77.4%	C	90.0-93.9%	A-	70.0-72.4%	C-	87.5-89.9%	B+	67.5-69.9%	D+	82.5-87.4%	B	60.0-67.4%	D	80.0-82.4%	B-	<60%	F	77.5-79.9%	C+		
>= 94%	A	72.5-77.4%	C																								
90.0-93.9%	A-	70.0-72.4%	C-																								
87.5-89.9%	B+	67.5-69.9%	D+																								
82.5-87.4%	B	60.0-67.4%	D																								
80.0-82.4%	B-	<60%	F																								
77.5-79.9%	C+																										

Learning Outcomes:

The learning outcomes for this course support the Computer Science and Information Systems learning outcomes as well as those of the College of Business and Economics:

Course Objectives	CSIS Learning outcomes	CBE Learning outcomes
Students will understand basic data structures.	Graduates will have knowledge of the latest programming methodologies.	Demonstrate Technology skills
Students will learn to design, implement and analyze programs using data structures.	Graduates will know the latest software engineering and design methods.	
	Graduates will know how to analyze, design and develop scalable software applications.	
Students will have the necessary skills to work in groups	Graduates will know how to work in teams with people with different backgrounds	Teamwork skills

Electronics

Turn off cell-phones and laptops before class. Texting, IMing, surfing, and use of Facebook or Twitter in class are unacceptable.

UW-RF has a policy on acceptable uses of University computers (applies to SH 218); see: <http://www.uwrf.edu/DOTS/AUP.cfm>

Attendance
policy:

You are responsible for attending class lectures. There are no make-up quizzes or labs. In general, you are responsible for catching up with the rest of the class if you miss a class. This means you should talk to your classmates and get copies of any handouts. If you know in advance that you may have to miss a class, please let me know as soon as possible.

Disability
Information and
Accommodations:

UW-RF welcomes students with disabilities into the University's educational programs and activities. Students with disability-related needs for reasonable accommodations should contact the Disability Services Office, 102 Davee Library or call 715-425-3531. Before final decisions can be made about allowing accommodation, students must provide sufficient documentation; see <http://www.uwrf.edu/AbilityServices/Index.cfm> for further details.

Software:

We will use a Linux system with g++ (v. 4.4.3+) to compile our C++ programs. There are [many](#) C++ compilers; we use the [GNU compiler](#) because it conforms to the ANSI C++ standards. We will use Eclipse with subversion.

For labs and assignments: Programs you submit should compile with the Gnu C++ compiler and should run on a Linux system.

Class Schedule

Class meetings	Text chapter/sections and topic	Labs, Assign
Jan. 28, 30, Feb 1	Eclipse, Savitch ch. 1: C++ Basics, 2: Control Flow	Lab #1
Feb. 4, 6, 8	Savitch ch. 3: Functions, 4: Overloading	Lab #2
Feb. 11, 13, 15	Savitch ch. 5: Arrays, 6: Structs and Classes, 7: Constructors	Lab #3
Feb. 18, 20, 22	Recursion I (Savitch ch. 13) ch 8: Operator Overloading, Friends, References	Lab #4
Feb. 25, 27, Mar 1	Savitch ch. 9: Strings, 10: Pointers, Dyn Arrays	Assign #1
March 4, 6, 8	Recursion II (Savitch ch. 13), ch. 11: Namespaces, 12: Streams	Lab #5
March 11, 13, 15	Savitch ch. 14: Inheritance, 15: Polymorphism, 16: Templates	Lab #6
March. 18, 20, 22	Savitch ch 17: Linked Lists	Assign #2
S P R I N G B R E A K		
Apr. 1, 3, 5	Savitch ch. 17: Linked Lists	Lab #7
Apr. 8, 10, 12	Savitch ch. 17: Stacks, Queues	Lab #8
Apr. 15, 17, 19	Savitch ch. 17: Hash Tables	Assign #3
Apr. 22, 24, 26	Savitch ch. 17: Binary Trees	Lab #9
Apr. 29, May 1, 3	Savitch ch. 17: Graphs	Lab #10
May. 6, 8, 10	Savitch ch. 17: Graphs	Assign #4
Final exam: SH 218 Thursday, May 16 @ 10:15 am - 12:15 pm		

Important dates:

Last day to drop a class without a 'W' noted on transcript:
Spring Break
Last day to drop a class
Last Day of Classes

February 15th
March 25th-31st
April 12th
May 10th

Important Contacts:

Ellen Schultz (Adviser, College of Business and Economics) SH Rm **124D**

Dr. Hossein Najafi (Chair of Dept of Computer Science and Information Systems) SH Rm **212C**

Course
Expectations:

This course is a team effort with the instructor and your classmates. Work in teams and help other class members learn. Form study groups. Ask questions in class and during office hours – online and in person.

Be prepared to spend a great deal of time outside class for this course. You are expected to

read the textbook and the material in D2L in advance of the lecture according to the class schedule above. Reading ahead will lower the amount of time required to get your programs working. If you are having difficulties understanding the lectures, assignments or exams, I will be glad to work with you outside class.

The labs in this course are meant to be started a week before they are due – they will usually require more than an hour's worth of work.

The online quizzes will have questions based on the readings.

Periodically the lecturer will call on you in class to answer questions regarding the readings. At the discretion of the lecturer, a good answer may substitute for an in-class summary.

The in-class summaries (worth 7.5% of your grade) are short essays you should write summarizing the the main points of the lectures.

This course involves a lot of programming. Most programs seldom work on the very first try.

- When your program does not work, use your time wisely to fix them.
- You learn more when you fix your own programs.
Every bug and every error is a learning opportunity – the first time you see it, you learn that it exists, the second time you see the same error/bug, you can learn how to fix it; ideally you will never see it again but if you do, you know what to do.
- If you have a program that is not working and you want some help, you can ask me in class or during office hours. You can also use the D2L online discussion forum.
Emailing me programs is **not productive** use of your time or mine – I cannot guarantee a timely reply because I have a limited amount of time for answering emails; see me in person.
- Before you ask (the instructor or your classmates) for help with a program, you will need to have done two things on your own:
First, make sure your program is **free of compile errors**.
If you have compile errors, **isolate** the cause of each error. The best way to do this is to “comment out” code that produces compile errors and then uncomment the good code. Ask for help in the D2L forums or <http://www.piazza.com/> (if there are enough students interested).
Second, use the **debugger** to track down bugs.
Insert comments in your code indicating what does not seem to work.
The more you can explain how your code works, the more people would be willing to listen to you and help you try to debug it.
Show us how far you got with tracking down bugs using the debugger.
- Learning how to debug programs is an extremely valuable skill.

Group work: when you work in a group, **you are 100% responsible** for doing **all the work** in an assignment or lab and for **turning in the work on time**.

If you are not satisfied with the work of others in your group, you **always** have the option of sending me your work (use the D2L one-on-one discussion forum) to be graded separately from your group.

This course is a team effort with the instructor and your classmates. Work in teams and help other class members learn. Form study groups.

All members of the UW-River Falls academic community are expected to:

- Prepare thoroughly
- Meet obligations punctually
- Participate fully and constructively

- Display appropriate courtesy
- Provide constructive feedback
- Respect confidentiality
- Challenge each other intellectually.

If you are not sure what questions to ask to get help, consider getting a tutor at the Academic Success Center in the Library.

Assignments:

I encourage teamwork for the **planning stage** (the design part) of all assignments unless I state otherwise. Discuss your approach to solving problems with classmates beforehand. Once you start writing the actual programs you intend to submit as your own work, **all work should be your own**. Once started, do not share your work with others until they have submitted their own work.

All programming assignment class files should have a header that looks like:

```

/*****
* Title:      Example Class
* Description: CS 237 Assignment #2
* Author:     Your Name Here
*             your.n.here@uwrf.edu
* Created:    February 21st 2013
* Modified:   February 23rd 2013
* Team:      Team Mate
*****/

```

Your code must be well documented. Up to half the points for an assignment may be reserved for documentation of working code. Turn in, using the Dropbox in D2L, a **single jar file** that contains the entire C++ program source code (use Eclipse).

Assignments are due as noted. Late assignments are not accepted.

If you want to use a code fragment that was written by someone else, you have to get that person's permission first as well as my permission before submission and, in addition, you will have to include a full citation in your code documentation. The easiest way to do the assignments is to put in the time to do it yourself.

Keep your graded assignments and quizzes until the end of the term in case of regrading or re-tabulation.

Academic Integrity:

I encourage teamwork for the **planning stage** (the design) of all assignments unless I state otherwise. Discuss your approach to solving problems with classmates beforehand. Once you start writing the actual programs you intend to submit as your own work, **all work should be your own**.

Once completed, do not share your work with others until they have submitted their own work. If someone turns in an assignment that is identical to yours, both of you are culpable.

UW-River Falls is a place of learning. Our main objective is to succeed academically so we are well prepared for a successful career.

Honesty is a greater long-term asset than good grades.

Any academic dishonesty in this course will be grounds for a failing grade (F) in the course. Academic dishonesty includes plagiarism and falsely representing the work of someone else as your own. If you are unsure of whether an action constitutes plagiarism, contact me. Also see: <http://www.uwrf.edu/policies/>