

## CSCI 2100: Data Structures

### Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
1.1	Catalog Description . . . . .	2
1.2	Prerequisites . . . . .	2
1.3	Detailed Description . . . . .	2
<b>2</b>	<b>Course Administration</b>	<b>3</b>
2.1	The Staff . . . . .	3
2.2	Class Meetings . . . . .	3
2.3	Textbook . . . . .	3
<b>3</b>	<b>Online Resources</b>	<b>4</b>
3.1	CSCI 2100 Web Page: <a href="http://cs.slu.edu/~goldwasser/2100/">cs.slu.edu/~goldwasser/2100/</a> . . . . .	4
3.2	Email with Instructor . . . . .	4
3.3	Electronic Assignment Submission . . . . .	4
<b>4</b>	<b>Graded Work</b>	<b>4</b>
4.1	Homework Assignments (15%) . . . . .	4
4.2	Lab Assignments (10%) . . . . .	5
4.3	Programming Assignments (35%) . . . . .	5
4.4	Exams (40%) . . . . .	5
4.5	Extra Credit . . . . .	5
4.6	Course Grades . . . . .	6
4.7	Academic Integrity . . . . .	6
4.8	Late Policies . . . . .	7
<b>5</b>	<b>Additional Information</b>	<b>8</b>
5.1	Supporting Student Success . . . . .	8
5.2	Tutoring Resources at SLU . . . . .	8
5.3	Computing Resources at SLU . . . . .	8

# 1 Overview

## 1.1 Catalog Description

“The design, implementation and use of data structures. Principles of abstraction, encapsulation and modularity to guide in the creation of robust, adaptable, reusable and efficient structures. Specific data types to include stacks, queues, lists, priority queues, dictionaries, trees and graphs.”

## 1.2 Prerequisites

The official prerequisite is a C- or better in CSCI 1300 and a passing grade or concurrent enrollment in MATH 1660 or equivalent.

## 1.3 Detailed Description

This is an undergraduate course that is a follow-up to an introductory programming course. The goal of that first course can be viewed as gaining the ability to write computer programs for accomplishing tasks. The key in that course was that programs must work *correctly* for the task at hand. In this course, we will focus on the fact that there are actually many different ways to correctly solve a particular problem and that these different methods may vary greatly in *efficiency* and *adaptability*.

What we will find is that writing a “good” program is not as simple as starting with a correct program and then improving it. Instead, the quality of a program is most determined by key thoughts and decisions that take place long before the first line of code might be written. Relevant issues include the choice of data structures that are used to organize all of the information within a program, and the structure and level of generality which allow components of a program to be better maintained, reused, and adapted.

In hand with this emphasis on writing good programs and designing useful data structures, the course will reinforce aspects of object-oriented programming using the **C++ programming language**. Since the majority of students will enter this course having used Python in CSCI 1300, we do not assume any familiarity with C++ at the onset of this course. In the early parts of the course, we will draw comparisons between these two languages to help in the transition.

## 2 Course Administration

### 2.1 The Staff

Instructor: Dr. Michael Goldwasser  
Email: [goldwamh@slu.edu](mailto:goldwamh@slu.edu)  
Web: <http://cs.slu.edu/~goldwasser/>  
Office: Ritter Hall 355  
Telephone: (314) 977-7039  
Office hours: Tuesdays 3:00–4:00pm  
Wednesdays 2:00–3:00pm  
Fridays 12:00–1:00pm  
or by appointment

Please make sure to take advantage of office hours, as they offer a wonderful opportunity for individual interaction.

### 2.2 Class Meetings

#### The Lectures

The material will be presented in four weekly meetings, one of which will typically be used as a hands-on lab. Attendance is expected and class participation is most welcome. These meetings will offer learning opportunities that cannot be recreated purely from readings. Information on the lecture topic can often be found on the course schedule web page.

Time: Mon/Wed/Thu/Fri, 10:00–10:50am  
Place: Ritter Hall 115

### 2.3 Textbook

The required textbook for this course is:

Title: *Data Structures and Algorithms in C++ (Second Edition)*  
Authors: Michael T. Goodrich, Roberto Tamassia, and David M. Mount  
Publisher: John Wiley & Sons, 2011  
ISBN: 978-0-470-38327-8  
Website: <http://www.wiley.com/college/goodrich>

For the beginning of the semester, we will also rely on the document *A Transition Guide from Python 2.x to C++*, available on the course web page.

## 3 Online Resources

### 3.1 CSCI 2100 Web Page: [cs.slu.edu/~goldwasser/2100/](http://cs.slu.edu/~goldwasser/2100/)

With the exception of the first day's printed handouts, most of the information for this course will be distributed only by means of the course web page. This web site will contain all assignments, a schedule of lectures, detailed lecture notes and links to many other sources of information. The web page contains some information (e.g. solutions, submitted assignments, individual grades) that is more sensitive and therefore which will be available to students in the class only after they have identified themselves properly. To gain access to these parts of the web page, a student must first complete an online questionnaire, creating a unique identity and password.

### 3.2 Email with Instructor

Face-to-face contact in class and in office hours is most desirable. Yet email is a convenient form of communication as well. I try to respond to email promptly, including at least once each evening when possible. If your question involves your progress on a current programming assignment, my response will be more informative if you can point out the specific problem you have encountered, and if I am able to see all of your source code. Therefore I strongly suggest that you either attach all relevant files to the email or submit preliminary versions of such files through our online system.

### 3.3 Electronic Assignment Submission

Most assignments for this course must be submitted electronically. Details of the submission procedure are given on the course web page.

## 4 Graded Work

### 4.1 Homework Assignments (15%)

We expect there to be a total of 8–10 written homework assignments during the course, each of which involves some amount of work on a computer. At the end of the semester, **we will throw away your lowest of the homework grades**, with the remaining assignments contributing equally to this portion of the grade.

These will generally be due at the beginning of class on the assigned due date. The problems will consist of short questions reinforcing the material in the lectures, readings and programs. Each homework will contain one or more practice problems that are not to be turned in and which can be discussed freely between classmates. The problems that are to be submitted for a grade, however, must be done entirely individually. A more complete explanation of our policy towards Academic Integrity is given in Section 4.7.

## 4.2 Lab Assignments (10%)

Most Thursdays will be devoted to a hands-on “lab” problem. Each will have a formal “pre-lab” exercise that must be submitted individually at the beginning of class that Thursday. The rest of the lab will be completed in pairs, ideally during the Thursday class period. However, we will allow for the lab (but not the prelab) to be submitted anytime by 11:59pm on the subsequent Friday.

The labs will be graded on a simple 3-point scale. One point is awarded for successfully submit the prelab exercise. A second point is awarded for any sincere attempt at the lab activity, and a third point is awarded for *successful* completion of the lab activity. At the end of the semester, **we will ignore your lowest of the lab grades**, with the remaining scores contributing equally to this portion of the grade.

## 4.3 Programming Assignments (35%)

We expect there to be a total of 8–10 programming assignments during the course. At the end of the semester, **we will ignore your lowest of the program grades**, with the remaining programs contributing equally to this portion of the grade.

On certain assignments, you will be required to work individually; on others you will be allowed to work in pairs. Early in the course you will have approximately one week for each assignment; later in the course you will often have two weeks. The assignments will be submitted electronically and generally due at 11:59pm on the assigned due date.

## 4.4 Exams (40%)

- **First Exam (10%)**, Thursday, 1 October 2015, 10:00–10:50am
- **Second Exam (10%)**, Thursday, 5 November 2015, 10:00–10:50am
- **Final Exam (20%)**, Monday, 14 December 2015, 12:00–1:50pm

## 4.5 Extra Credit

Both homework and programming assignments will generally include a small extra credit challenge. Please notice, however, that the actual extra credit given for these challenges is relatively insignificant. Students who are seriously concerned about improving their overall grade would be best advise to focus all efforts on doing as well as possible on the required work and in preparing for exams.

Our true reason for including these opportunities is to provide some fun and encouragement for students who wish to dig a bit deeper than was required in an assignment. For those students, the chosen extra credit challenges provide a good next step.

## 4.6 Course Grades

Letter grades will be based on each student's overall percentage of awarded points according to the following formula.

- Student percentage above 90% will result in a grade of A or better.
- Student percentage above 87% will result in a grade of A- or better.
- Student percentage above 83% will result in a grade of B+ or better.
- Student percentage above 80% will result in a grade of B or better.
- Student percentage above 77% will result in a grade of B- or better.
- Student percentage above 73% will result in a grade of C+ or better.
- Student percentage above 60% will result in a grade of C or better.
- Student percentage above 67% will result in a grade of C- or better.
- Student percentage above 60% will result in a grade of D or better.
- Student percentage below 60% will result in a grade of F.

Any modification to this scale at the end of the year will be *in favor of the students*. That is we may later decide to award an A to a student who is slightly below the above cutoff, but we certainly will not deny an A from someone who is above the cutoff.

## 4.7 Academic Integrity

Academic integrity is honesty, truthful and responsible conduct in all academic endeavors. The mission of Saint Louis University is “the pursuit of truth for the greater glory of God and for the service of humanity.” Accordingly, all acts of falsehood demean and compromise the corporate endeavors of teaching, research, health care, and community service via which SLU embodies its mission. The University strives to prepare students for lives of personal and professional integrity, and therefore regards all breaches of academic integrity as matters of serious concern.

The governing University-level Academic Integrity Policy can be accessed on the Provost's Office website. A more detailed policy statement is given by the College of Arts & Science ([www.slu.edu/colleges/AS/academic\\_honesty.html](http://www.slu.edu/colleges/AS/academic_honesty.html)), also applying to this course. In addition to those general statements, we wish to discuss our policy in the context of this course. When it comes to learning and understanding the **general course material**, you may certainly use other reference materials and you may have discussions with other students in this class or other people from outside of this class. This openness pertains to material from the text, practice problems, general syntax and use of the C++ language or other computing tools.

However, when it comes to **work that is submitted for this course**, you are not to use or to search for any direct or indirect assistance from unauthorized sources, including but not limited to:

- other students in this class
- past students, whether from this school or other schools

- other acquaintances
- other texts or books
- online information other than that referenced by course materials

Acceptable sources of information include consultations with the instructor, teaching assistants, or members of organized tutoring centers on campus, as well as any materials explicitly authorized in an assignment. Even in these cases, if you receive significant help you should make sure to document both the source of the help as well as the extent.

On certain programming assignments, we will explicitly allow students to work in pairs. In this case, conversations between partners is both permissible and required. Furthermore, both students are expected to contribute significantly to the development of the submitted work. It is unethical to allow a partner to “sign on” to a submission if that partner did not significantly contribute to the work.

Any violations of these policies will be dealt with seriously. Penalties will apply as well to a student who is aiding another student. Any such violations will result in a minimum penalty of a zero on the given assignment that cannot be dropped, and severe or repeated violations will result in an immediate failing grade in the course. Furthermore all incidents will be reported in writing to the Department and/or the Dean, as per the College procedure.

## 4.8 Late Policies

All exams must be taken promptly at the required time. Requests for rescheduling an exam will only be considered if the request is made prior to the start of the exam, or else in an “emergency” situation with appropriate documentation.

For both written and programming assignments, we wish to allow students to continue to work comfortably beyond the official deadline when a little more time will result in more progress, while at the same time discourage students from falling significantly behind pace and jeopardizing their success on future assignments. Our solution is the following exponentially decaying late formula (some have suggested that we should offer extra credit to anyone who fully understands this formula).

We will consider an assignment submission “complete” when any part of the assignment is last submitted or modified. Any assignment that is not complete promptly by its due date and time will be assessed a penalty based on the formula  $S = R \cdot e^{-h/173}$ , where  $S$  is the grade given,  $R$  is the grade the work would have received had it been turned in on time, and  $h$  is the amount of time (in hours or fractions thereof) that the work was late. Examples:

- work turned in 1 hour late receives over 99.6% of its original credit
- work turned in 5 hours late receives over 97% credit
- work turned in one full day late receives less than 88%
- work turned in two full days late receives less than 76%
- work turned in five days late receives less than 50%

The above policies will be waived only in an “emergency” situation with appropriate documentation.

## 5 Additional Information

### 5.1 Supporting Student Success

In recognition that people learn in a variety of ways and that learning is influenced by multiple factors (e.g., prior experience, study skills, learning disability), resources to support student success are available on campus. Students who think they might benefit from these resources can find out more about:

- Course-level support in the remainder of this section or by asking the instructor
- University-level support (e.g., tutoring/writing services, Disability Services) by visiting the Student Success Center (BSC 331) or by going to [www.slu.edu/success](http://www.slu.edu/success).

Students with a documented disability who wish to request academic accommodations are encouraged to contact Disability Services at 314-977-3484 or visit the Student Success Center. Confidentiality will be observed in all inquiries.

### 5.2 Tutoring Resources at SLU

Our department employees many junior/senior computer science majors to help out in our department labs. Those students are also available to provide assistance with course materials at such times.

Our department web page maintains a current list of the available times and locations at [cs.slu.edu/undergrad-cs/lab-hours](http://cs.slu.edu/undergrad-cs/lab-hours).

As stated in the Academic Integrity policy of Section 4.7, these tutors are an acceptable resource for help, yet you should still document both the source of the help as well as the extent, if significant.

### 5.3 Computing Resources at SLU

Our department runs a computer server named `turing` that serves as the primary computing environment for this course. If you do not already have an account on this machine, please let us know and we will create a new one.

See [cs.slu.edu/computing-resources](http://cs.slu.edu/computing-resources) for further documentation regarding use of `turing`'s facilities.

**Please note:** you are not explicitly required to use `turing` as your computing platform, it is simply the only platform what we will officially support. If you wish to use another platform that offers you sufficient support for completing your assignments, please feel free to do so.