

# CSCI 1300

## Introduction to Object-Oriented Programming

### Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
1.1	Catalog Description . . . . .	2
1.2	Prerequisites . . . . .	2
1.3	Student Learning Outcomes . . . . .	2
<b>2</b>	<b>Outline of Topics</b>	<b>2</b>
<b>3</b>	<b>Course Administration</b>	<b>3</b>
3.1	Website: <a href="http://cs.slu.edu/~goldwasser/1300">cs.slu.edu/~goldwasser/1300</a> . . . . .	3
3.2	Class Meetings . . . . .	3
3.3	Textbook . . . . .	3
3.4	The Staff . . . . .	3
<b>4</b>	<b>Flipped Classroom</b>	<b>4</b>
<b>5</b>	<b>Graded Work</b>	<b>4</b>
5.1	Reading Assignments (10%) . . . . .	4
5.2	Homework Assignments (10%) . . . . .	5
5.3	Quizzes (10%) . . . . .	5
5.4	Programming Assignments (30%) . . . . .	5
5.5	Exams (40%) . . . . .	5
5.6	Extra Credit . . . . .	5
5.7	Course Grades . . . . .	7
5.8	Late Policies . . . . .	7
<b>6</b>	<b>Academic Integrity and Collaboration Policy</b>	<b>8</b>
6.1	Academic Integrity . . . . .	8
6.2	Collaboration Policy . . . . .	9
<b>7</b>	<b>Additional Information</b>	<b>9</b>
7.1	Perusall Reading Assignments . . . . .	9
7.2	Electronic Submission via Git . . . . .	9
7.3	Title IX Statement . . . . .	10
7.4	Supporting Student Success . . . . .	10
7.5	Disability Services . . . . .	10

# 1 Overview

## 1.1 Catalog Description

An introduction to computer programming based upon early coverage of object-oriented principles such as classes, methods, inheritance and polymorphism, together with treatment of traditional flow of control structures. Good software development practices will also be established, including issues of design, documentation, and testing.

## 1.2 Prerequisites

MATH 1200 (College Algebra) or equivalent, and a C- or better in one of CSCI 1010 through 1090, or equivalent programming experience.

## 1.3 Student Learning Outcomes

After successfully completing this course, students will be able to:

1. Accurately predict the behavior of small pieces of code authored by others, including use of control structures and interacting objects.
2. Make use of data types and control structures in order to implement high-level behaviors.
3. Write, debug, and document a well-structured program, of at least 100 lines of code, that functions in accordance with desired specifications.
4. Make use of objects from a class defined by someone else (such as built-in string and list classes, or from other language APIs)
5. Implement a user-defined class based upon given specifications, and make use of inheritance to design a subclass of another
6. Demonstrate an emergent knowledge of recursion through simulation of existing code or implementation of simple recursive functions.

# 2 Outline of Topics

Variables and Expressions

User Input/Output

Basics of Object-Orientation and Using Objects

Conditional Statements

Loops

User-defined Functions

User-defined Classes

Good Software Practices

Inheritance

Container Classes

Recursion

## 3 Course Administration

### 3.1 Website: [cs.slu.edu/~goldwasser/1300](http://cs.slu.edu/~goldwasser/1300)

### 3.2 Class Meetings

The class has four weekly meetings. We will be employing a flipped classroom model for most of the semester and attendance and class participation is expected. These meetings will offer learning opportunities that cannot be re-created purely from readings. Information on the lecture topic can be found on the course schedule web page.

Time: Mon/Tue/Wed//Fri, 10:00–10:50am

Place: Ritter Hall 115

### 3.3 Textbook

We will be creating custom readings this semester, which will be released through an online system named Perusall. More information is given in section 7.1.

### 3.4 The Staff

**Instructor:** Dr. Michael Goldwasser  
**Email:** [michael.goldwasser@slu.edu](mailto:michael.goldwasser@slu.edu)  
**Web:** [cs.slu.edu/~goldwasser/](http://cs.slu.edu/~goldwasser/)  
**Office:** Ritter Hall 335  
**Phone:** (314) 977-7039  
**Office hours:** Monday 2:00–3:00pm  
                  Tuesdays 11:00–11:50am  
                  Fridays 9:00–9:50am  
                  or by appointment

Face-to-face contact in class and in office hours is the most desirable means for individual attention and mentorship. With that said, email is another convenient form of communication. I try to respond to email promptly, including at least once each evening. However, if you have a question involving progress on a current programming assignment, we ask that you use the electronic submission system which allows you to submit work in progress for examination, and an issue tracker that will allow for online discussion between student and instructor about those materials.

In addition to the instructor, we will employ several peer assistants, both to provide guidance during in-class activities, and to hold drop-in tutoring hours at various times during the week.

## 4 Flipped Classroom

We will employ a “flipped classroom” approach to the semester, constructed around what we will term units, with most units matching a single chapter of the textbook and involving two face-to-face class meetings (although for some units we might adjust this flow somewhat). The typical flow of the unit is as follows:

	Before Class	During Class	After Class
First Day	Two days prior, instructor posts reading on perusall.  By 8:00am of class day, students must have completed the reading and sufficiently interacted with the material through perusall.	The instructor will review the reading assignment with the class, paying particularly to areas of concern/interest that were identified from the reading assignment. There will also be a question/answer period and a preview of the day to follow.	Homework exercises to review the unit are available; due next class (individually).
Second Day	Homework Exercises due at 10:00am.	Pairs randomly chosen.  40 minutes of practice, exercises, challenges for pairs.  10 minute quiz, completed and submitted as a pair.	Get started on next unit, and any out-of-class programming assignments.

## 5 Graded Work

### 5.1 Reading Assignments (10%)

There will be a series of reading assignments that require student to prepare for class by reading and annotating chapters of the text through an online system Named Perusall (see Section 7.1). That system will lock the assignment at 8:00am on the days that the readings will be discussed in class; we will not allow late assignments.

We expect there will be between 20–24 reading assignment during the semester. Perusal “scores” student engagement on a 0–3 scale. We will calibrate by considering a score of 2 to be full credit (and a score of 3 to be extra credit). To provide some additional forgiveness for missed or poor assignments, **we will drop the lowest 4 reading assignment grades at the end of the semester**, and the remaining grades will contribute equally to this portion of the grade.

## 5.2 Homework Assignments (10%)

For each class day that is designated as a hands-on day, there will be a pen-and-paper homework set containing exercises from the relevant reading. This homework will be due at the *beginning* of the class meeting and we will not accept late homeworks.

We expect there will be 25 to 30 such assignments during the semester. To provide some forgiveness for missed or poor assignments, **we will drop the lowest 5 homework assignment grades at the end of the semester**, and the remaining grades will contribute equally to this portion of the grade. All of the homeworks must be done independently, and in accordance with the policy on Academic Integrity as given in Section 6.1.

## 5.3 Quizzes (10%)

Most hands-on days will end with a ten minute quiz, to be completed and submitted by a pair of students who have been working together during that class period. We expect there will be 25 to 30 such quizzes during the semester. To provide some forgiveness for missed or poor quizzes, **we will drop the lowest 5 quiz grades at the end of the semester**, and the remaining grades will contribute equally to this portion of the grade.

## 5.4 Programming Assignments (30%)

Beyond the in-class work, we expect there to be about 10 more significant programming assignments during the semester. At the end of the semester, **we will throw away the one lowest of the program grades**, with the remaining scores contributing equally to this portion of the grade. These assignments will be submitted electronically (details in Section 7.2), and due at 11:59pm on the assigned due date. See Section 5.8 for the policy on late submission. On certain assignments, you will be required to work individually; on others you will be allowed to work in pairs. Please respect the policy on Academic Integrity as given in Section 6.1.

## 5.5 Exams (40%)

- **Midterm Exam (10%)**, Friday, 28 September 2018, 10:00–10:50am
- **Midterm Exam (10%)**, Friday, 2 November 2018, 10:00–10:50am
- **Final Exam (20%)**, Monday, 17 December 2018, **8:00–9:50am**

## 5.6 Extra Credit

Homework and programming assignments will sometimes include a small extra credit challenge. Please notice, however, that the actual extra credit given for these challenges is relatively insignificant. Students who are concerned about improving their overall

grade would be best advised to focus efforts on doing as well as possible on the required work and in preparing for exams.

Our real 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.

## 5.7 Course Grades

Letter grades will be based on each students 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 70% 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.

## 5.8 Late Policies

The timing of the various assignments, quizzes, and exams are orchestrated based around the classroom activities and so we will not allow any late submissions of reading assignments or homework assignments.

Quizzes and exams must be taken at the regularly schedule time unless advanced arrangements have been made for unavoidable conflicts or subsequently due to emergency situations with appropriate documentation

For the out-of-class 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 which 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%

## 6 Academic Integrity and Collaboration Policy

### 6.1 Academic Integrity

*Academic integrity is honest, 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 was adopted in Spring 2015, and can be accessed on the Provost’s Office website.

Additionally, each SLU College, School, and Center has adopted its own academic integrity policies, available on their respective websites. All SLU students are expected to know and abide by these policies, which detail definitions of violations, processes for reporting violations, sanctions, and appeals. Please direct questions about any facet of academic integrity to your faculty, the chair of the department of your academic program, or the Dean/Director of the College, School or Center in which your program is housed. Specific College of Arts and Sciences Academic Honesty Policies and Procedures may be found at: [www.slu.edu/arts-and-sciences/student-resources/academic-honesty.php](http://www.slu.edu/arts-and-sciences/student-resources/academic-honesty.php)

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 and practice problems.

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

- other texts, books, or solution manuals
- online information other than that referenced by course materials
- other students in this class (other than when collaboration is explicitly allowed, as described below)
- students or acquaintances who are not in this course

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.

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 which 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.

## 6.2 Collaboration Policy

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, all 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.

# 7 Additional Information

## 7.1 Perusall Reading Assignments

The reading assignments for this course will be managed using an online tool from [perusall.com](http://perusall.com). PDFs of the course materials will be uploaded to that site and available to students, and students are able (and required) to interact with the materials by marking up the reading with questions and comments, and viewing and responding to comments of other students. In order to participate:

- Students must create a new account at [perusall.com](http://perusall.com), signing in either with Facebook/Google/Twitter or newly created credentials.
- Students must join the course using the course code: **GOLDWASSER-73340**
- Students must complete the reading assignments by reading and annotating the reading chapters. A student’s grade for a reading assignment will be computed by perusall as follows. Each comment or question entered by a student is scored on a 0/1/2-point scale for below/meets/exceeds expectations, and the four highest-score comments will contribute to the overall grade for the reading assignment.

## 7.2 Electronic Submission via Git

To allow the student and instructor to exchange electronic files for programming assignments, we will rely on a version control system known as git, and a web-based system known as gitlab and available at [git.cs.slu.edu](http://git.cs.slu.edu). For further documentation of the use of this system, please see the course webpage.

### 7.3 Title IX Statement

Saint Louis University and its faculty are committed to supporting our students and seeking an environment that is free of bias, discrimination, and harassment. If you have encountered any form of sexual misconduct (e.g. sexual assault, sexual harassment, stalking, domestic or dating violence), we encourage you to report this to the University. If you speak with a faculty member about an incident of misconduct, that faculty member must notify SLU's Title IX coordinator, Anna R. Kratky (DuBourg Hall, room 36; [akratky@slu.edu](mailto:akratky@slu.edu); 314-977-3886) and share the basic fact of your experience with her. The Title IX coordinator will then be available to assist you in understanding all of your options and in connecting you with all possible resources on and off campus.

If you wish to speak with a confidential source, you may contact the counselors at the University Counseling Center at 314-977-TALK. To view SLU's sexual misconduct policy and for resources, please visit the following web address: [www.slu.edu/here4you](http://www.slu.edu/here4you).

### 7.4 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. The Student Success Center, a one-stop shop, which assists students with academic and career related services, is located in the Busch Student Center (Suite 331) and the School of Nursing (Suite 114). Students can visit [www.slu.edu/success](http://www.slu.edu/success) to learn more about:

- Course-level support (e.g., faculty member, departmental resources, etc.) by asking your course instructor.
- University-level support (e.g., tutoring services, university writing services, disability services, academic coaching, career services, and/or facets of curriculum planning).

### 7.5 Disability Services

Students with a documented disability who wish to request academic accommodations **must** contact Disability Services to discuss accommodation requests and eligibility requirements. Once successfully registered, the student also **must** notify the course instructor that they wish to access accommodations in the course.

Please contact Disability Services, located within the Student Success Center, at [Disability\\_services@slu.edu](mailto:Disability_services@slu.edu) or 314-977-3484 to schedule an appointment. Confidentiality will be observed in all inquiries. Once approved, information about academic accommodations will be shared with course instructors via email from Disability Services and viewed within Banner via the instructor's course roster.

Note: Students who do not have a documented disability but who think they may have one are encouraged to contact Disability Services.