# CSCI 1300/5001: Introduction to Object-Oriented Programming

## (Spring 2020, Dr. Michael Goldwasser)

## Catalog Course 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.

## 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.

## Outline of Topics

- Variables and Expressions
- User Input/Output
- Basics of Object-orientation
- Interacting with Objects
- Conditional Statements
- Loops
- User-defined Functions
- User-defined Classes
- Good Software Practices
- Inheritance
- Container Classes
- Recursion

## Contents

# Course Information

## Prerequisites

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

## Class Meetings

The class has four weekly meetings:

Time: Mon/Tue/Wed/Fri, 2:10–3:00pm
Place: Ritter Hall 115

We will be employing a flipped classroom model for most of the semester (see page 3) and attendance and class participation is expected. The grading policy (see page 6) provides relief for a limited number of absences.

## Staff

**Instructor:** Dr. Michael Goldwasser
Email: michael.goldwasser@slu.edu
Web: cs.slu.edu/~goldwasser/
Office: Ritter Hall 335
Phone: (314) 977-7039
Office hours: Mondays 9:00–10:00am
Tuesdays 3:00–4:00pm
Fridays 1:00–2:00pm
or by appointment

**Communication:** 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.

**Peer Assistants:** In addition to the instructor, we will employ peer assistants, both to provide support for in-class activities and to hold drop-in tutoring hours at various times during the week (see page 9).

## Textbook

We will be using a draft to a second edition of *Object-Oriented Programming in Python* by Michael Goldwasser and David Letscher. This draft will be available for free through our online course technologies (see page 4).

# Flipped Classroom

## Organization

Throughout the semester, we will employ a "flipped classroom" approach, constructed around a series of units, each centered around a chapter of the textbook. Typically, two class meetings will be devoted to an exploration of a unit, structured as follows. (Note: for some topics, we will choose to vary this structure to use one or three days.)

## Before "Day 1"

Typically, the instructor will post a reading assignment two days prior to the introduction of a new topic in class. **By 12:00pm of the designated "day 1" for a topic**, students are responsible for having:

- Engaged in the online reading
- Answered an online set of basic reinforcement questions
- Completed an online learning survey

## "Day 1" — Theory Day

During the first day engaging a new topic, the instructor will employ a variety of techniques that may include a combination of:

- Providing a mini-lecture on the most meaningful (of confusing) portions of the chapter
- Answering common questions raised by students
- Exploration of source code or live Python demos
- Leading group learning activities

## Between "Day 1" and "Day 2"

There will typically be an individual homework that consists of several questions that require deeper engagement in the new material. The homework will be released alongside the original reading, so that students can consider (and perhaps complete) the questions while doing the reading. However, the homework will be **due at 2:10pm at the beginning of "day 2"** so that students may use "day 1" to gain greater understanding of the new material, and to ask clarifying questions.

## "Day 2" — Hands-on Day

For most topics, the goal of the second class meeting will be to rapidly gain experience with new material in a programming environment. We will (loosely) employ the technique of "pair programming" having students work together in pairs each day (with those pairs randomly re-assigned for each such class period).

Within such a 50-minute meeting, we will typically provide 40 minutes of practice time, with an ample supply of programming challenges, many with embedded solutions, and none of which are to be submitted. In the final 10 minutes, a similar challenge problem will be provided as a formal quiz to be **completed and submitted by the pair of students**.

# Technologies

## Moodle Content Management System

The primary website for this course will be a Moodle course page, available at cs.slu.edu/moodle. Your username is your SLUNetID. You can reset the password the first time you login.

You should be checking the site daily for tasks that need to be completed prior to the next class and for assignments that follow up on in-class activities.

## Git Repositories for Submission of Software and Feedback

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. Your username for this system is your SLUNetID, and you may set your initial password by clicking the "Forgot my password" link the first time you login.

## Python 3 Programming Language

Python is an open-source programming language that is freely-available for all major computing platforms. Something to be aware of is that there are two major versions of Python currently in use (the so called Python 2.x branch and Python 3.x branch). **We will be using the 3.x line in this course** (and it is not backward compatible with the 2.x branch).

If you are using your account on our department's hopper system, Python is already installed and can be started by typing the command python3 in a console window (not to be confused with the command python which is the default 2.x branch). There is also a development environment for Python known as IDLE that is available by navigating the start menu to Applications → Development → IDLE3.

If you wish to run locally on your own computer, Python can be easily installed on all major computing platforms. Download a Python installer from www.python.org/download; **make sure to download the Python 3.x line** (Python 3.8.1 is the current release as of January 2020). Note that OSX comes pre-installed with Python, but only with the Python 2 distribution.

While it will be helpful to have the Python interpreter installed on your own computer, there are actually many very good web-based tools that would provide you with either an interactive interpreter or even ways to execute Python scripts through a browser. Those are great both for exploring, and perhaps even for completing assignments. In the end, what you will typically be submitting for programming assignments is your source code, so if you can develop that on a web-based platform, that's okay. Here are some sites that we can recommend:

- The official Python organization's website provides an interactive interpreter on their website at www.python.org/shell
- There is a wonderful web site named pythontutor.com that allows you to walk through the execution of a Python program, step-by-step, while providing a helpful visualization of your current workspace.
- The site repl.it/site/languages/python3 provides a full IDE for developing and executing Python code in the browser.

## The `cs1graphics` module

The graphics support for this class is provided by a module named `cs1graphics` developed here at SLU. It is not an official part of the Python distribution, so you will need to make sure to have it available on your system as well. If working on our department system, it is already available. If working on your own system, you will need to download a single file named `cs1graphics.py`. The most recent public release of this software is available at `www.cs1graphics.org`.

There are two ways to use the file on your own computer. One is to make sure to place the file in the same directory in which you are working when you start the Python interpreter. A better approach is to install the package by placing the file on your system in a folder typically named `site-packages` that is part of the Python distribution. If you have trouble finding the location of that package, please type the following command while within the Python interpreter.

```
>>> import sys; print([p for p in sys.path if 'site-packages' in p])
```

## Perusall Reading Platform

All of our course readings will be done using the website Perusall. You will not need a login for the site as all readings should be accessed directly from Moodle.

On Perusall you will be able to read the text, but also to highlight portions of the textbook and participate in discussion threads about the material. Once a thread is started on a selection, other students and the instructor are able to participate by making further comments, answering questions, or upvoting other people's questions or answers. Ways you might annotate the text include:

- If you don't understand a step in an example, highlight it and ask your classmates for assistance.
- Give your classmates tips that might help them understand a passage.
- If you see a connection in the reading to something done previous in the course point it out.
- Answer your classmates questions.

You might also do an initial reading a day in advance, and come back a bit later to view and respond to other student's comments. A Perusal reading assignment will **remain active until 12:00pm on the indicated due date** (typically the morning of "day 1" in our flipped model).

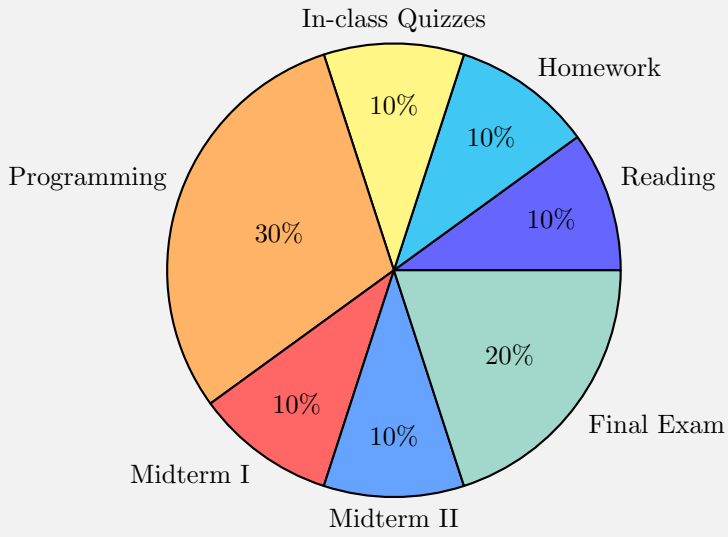
The goal of a reading is to gain initial exposure to fundamental concepts and techniques. We suggest you read each section and carefully work through all of the examples (ideally by trying things out yourself in Python). This initial exposure to the material is a critical step in the learning process and should not be undervalued.

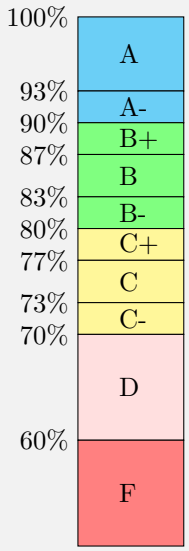
The Perusall system performs an automated evaluation of your reading engagement. We ask that you keep in mind that the true goal is your learning and not to get too caught up in its scoring system. The system is calibrated to score a student's engagement on a scale of zero to three, and we will consider a grade of two or higher as successful completion of the assignment. As discussed on page 6, we will also offer other ways to demonstrate your readiness for class.

With that said, factors that positively impact Perusall's scoring system include:

- Spending sufficient time reading.
- Spending time and effort distributed across the entire length of the reading.
- Making at least four high-quality annotations.
- Providing comments/questions that elicit other meaningful replies.

## Grading Policies




**Reading (10%)** A student's preparation before each "day 1" in our flipped model will be evaluated on a two-point scale, with *three* opportunities to achieve one of those two points:

| | |
|---|---|
| 1 point | Complete reading with score of 2 or higher in Perusall's system |
| 1 point | Score 75% or higher on the Moodle reading quiz |
| 1 point | Complete the learning survey in Moodle |

At the end of the semester, the lowest 20% of these grades will be dropped.

**Homework (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. At the end of the semester, the lowest 20% of these grades will be dropped.

All of the homeworks must be done independently, and in accordance with the policy on Academic Integrity as given on page 8.

**In-class 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. At the end of the semester, the lowest 20% of these grades will be dropped.

**Programming (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, the lowest of these grades will be dropped.

These assignments will be submitted electronically through git (details on page 4), and due at 11:59pm on the assigned due date. See page 7 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 on page 8.

**Midterms (10% each) and Final exam (20%)** Two midterm exams are tentatively scheduled for **February 14** and **March 27** and a final exam on **May 6**, 2:00–3:50pm.

Final grades will be assigned using the standard scale above. At the discretion of the instructor, grades may be curved yet only to increase a student's grade.

## 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.

## Late Policy

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 approprite 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%

## 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 through which SLU fulfills 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 full University-level Academic Integrity Policy can be found on the website of the Provost's Office.

Additionally, each SLU College, School, and Center has 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

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 (except 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.

## Collaboration Policy

On selected portions of assignments for this course, 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.

# Additional Information

## 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 assists students with academic-related services and is located in the Busch Student Center (Suite, 331). Students can visit www.slu.edu/life-at-slu/student-success-center/ to learn more about tutoring services, university writing services, disability services, and academic coaching.

Furthermore, the Department of Computer Science provides drop-in tutoring support for our introductory classes as well as use of our department's Linux computing systems. Tutoring hours are held in the Linux Lab (Ritter Hall 117) and no appointments are necessary. The schedule for available hours can be found at cs.slu.edu/resources/tutoring.

## Disability Accommodations

Students with a documented disability who wish to request academic accommodations must formally register their disability with the University. Once successfully registered, students also must notify their course instructor that they wish to use their approved accommodations in the course.

Please contact Disability Services to schedule an appointment to discuss accommodation requests and eligibility requirements. Most students on the St. Louis campus will contact Disability Services, located in the Student Success Center and available by email at Disability_services@slu.edu or by phone at 314-977-3484. Once approved, information about a student's eligibility for academic accommodations will be shared with course instructors by email from Disability Services and within the instructor's official course roster. Students who do not have a documented disability but who think they may have one also are encouraged to contact to Disability Services. Confidentiality will be observed in all inquiries.

## Title IX

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 that involves a Title IX matter, **that faculty member must notify SLU's Title IX coordinator (or that person's equivalent on your campus) and share the basic facts of your experience.** This is true even if you ask the faculty member not to disclose the incident. The Title IX contact 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.

For most students on the St. Louis campus, the appropriate contact is Anna R. Kratky (DuBourg Hall, room 36; anna.kratky@slu.edu; 314-977-3886). 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 addresses: www.slu.edu/here4you and www.slu.edu/general-counsel.