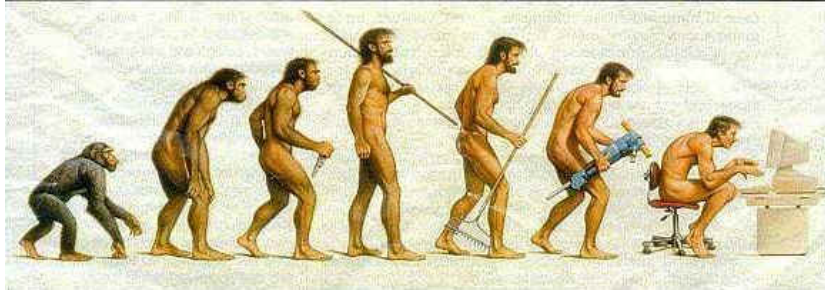


# CS 10, Winter 2014

## Problem Solving via Object Oriented Programming



---

### Course description

**ORC.** Motivated by problems that arise in a variety of disciplines, this course examines concepts and develops skills in solving computational problems. Topics covered include *abstraction* (how to hide details), *modularity* (how to decompose problems), *data structures* (how to efficiently organize data), and *algorithms* (procedures for solving problems). Laboratory assignments are implemented using object-oriented programming techniques.

### Prerequisite

CS 1, CS 5, or Engineering Sciences 20, or placement through AP or local placement exam

---

### Who, when, where

#### Instructor

[Chris Bailey-Kellogg](#) | 250 Sudikoff | office hours: M 1-2:30, F 11:30-1, and by appointment

#### Teaching Assistant

[Travis Peters](#) | 205 Sudikoff | office hours: M 3-4:30, Th 1:30-3

[Haider Syed](#) | 205 Sudikoff | office hours: Th 3-4:30, F 2:30-4

#### Section leaders

Ryan Amos, Mahina Kaholokula, Naho Kitade, Jae Heon Lee, Patty Neckowicz, Daniel Shanker, Xinran Xiao, Richard Yu, Joanne Zhao, Joy Zhong

#### Help

We will be using [Canvas](#) / Piazza. See postings there for more info.

#### Lectures

10-hour | MWF 10:00-11:05; Th 12:00-12:50 | Cook room in Murdough

The x-hour will primarily be used as an optional, informal, interactive session of working through examples together. It may sometimes be used to make up for missed classes.

#### Section meetings

These are weekly small-group meetings with section leaders to review lecture material, discuss questions, go over homeworks, etc. Groups and times will be posted on Canvas.

#### Lab

Help will be available in 003 Sudikoff Sun afternoon 2-5, and Sun, Mon, and Tue evenings 7-10

Students enrolled in CS 10 should have automatic access via ID card to Sudikoff and the instructional labs; for assistance see Holly Stearne in the Sudikoff Office, and email or see me if there are problems.

#### Announcements

Monitor [Canvas](#) for periodic course-wide announcements.

#### Textbook

Required: [Data Structures & Algorithms in Java \(fifth edition\)](#), by Michael Goodrich and Roberto Tamassia. Note that a 6th edition is now out, but wasn't when we had to place the textbook order, so I will be sticking with the 5th edition.

---

### Coursework

All homeworks (short assignments and problem sets) for this class are to be submitted electronically via [Canvas](#) by the start of class (10am). Even when an assignment has some written exercises, you are required to either type in a file or scan your written work and submit it electronically. To submit output from your program, submit a copy-pasted file in plain text format and/or a screenshot, as appropriate. For plain text, you can use a program like TextEdit, NotePad, or Emacs, or even Word, but be sure to save as plain text. For a screen shot, you can use Preview on Mac (under the "File" menu) or the PrntScrn button on Windows.

Sample solutions will be made available on Canvas.

#### Short assignments (10%)

Practice and feedback to aid your understanding of the basic course material.

### *Late policy*

Due via Canvas by the start of class. No credit for late submissions.

### *Grading*

2 (correct and good), 1 (needs work), or 0 (nothing of substance).

A solution receiving a 1 may be revised and resubmitted *once*, before the next regular class period after it is returned, for a possible upgrade to 2.

### *Problem sets (50%)*

A mixture of written and in-depth programming exercises, challenging you to use the ideas we study in class to solve new problems.

### *Late policy*

Due via Canvas by the start of class. Penalties: < 8 hours: 10%; < 24 hours: 20%; < 48 hours: 40%; more: no credit.

You are allowed at most one late submission (up to 48 hours) with no penalty; no excuse required. Indicate in your submission that you are electing to use your free pass; no undoing the choice.

### *Grading*

Specific grading rubrics will be provided for each problem set, covering *correctness* (solving the assigned problem), *structure* (organization, use of techniques covered, efficiency), *style* (readability of code, clarity of documentation), and *testing* (your demonstration of correctness).

### *Exams (40%)*

A midterm and a final exam, each worth 20%.

### *Extra credit*

Some homeworks have specific extra-credit problems; you may suggest other extensions for possible extra credit; any exceptionally clever, creative, or insightful work may likewise be awarded extra credit points. As its name suggests, extra credit is always optional, and you should never feel that you have to do extra credit problems. Extra credit points are recorded separately from other grades, and may be used to help make borderline letter grade assignments at the end of the term (those with *substantial* extra credit would get the higher grade if their grade is on the borderline). Extra credit points can only help, never hurt, your final grade, regardless how much or how little extra credit you or your classmates choose to do. However, you should not view extra credit as a substitute for doing good and thorough work on your assignments.

---

## ***Collaboration and honor code***

Dartmouth's honor code applies to this course, and academic misconduct policies will be strictly enforced. If you have questions, ask!

### *Output*

The output that you submit (screen shot/copy-pasted text) must be unaltered/unedited from what is produced by the program that you submit. If you make any change to the code, no matter how insignificant you think it might be, rerun your program to produce new output!

### *Piazza*

As described above, we will use Piazza as a shared help system. Your public posts (questions and replies) must not reveal solutions (or even partial solutions) to the problems.

### *Sample solutions*

You may not refer to sample solutions in revising your code for resubmission or late submission.

### *Partners*

Much of the learning in this course comes from doing the programming exercises. On some problem sets, you may work jointly with *one* other person, if so stated. No more than two people may work together on a given problem set. If you choose to work with someone else, you and your partner must submit a single joint assignment with both names on it, and you must work with the same person for the entire assignment (you cannot work with one person for some parts of an assignment and a different person for other parts).

On short assignments, you must work *alone* unless otherwise stated in the assignment.

### *Attribution*

Under no circumstances may you hand in work done with (or by) someone else under your own name. If you have any doubt, credit any person(s) (other than course staff) from whom you got help. Your code should *never* be shared with anyone, other than your partner (if you are working in a pair). You would be amazed at how easy it is to tell when people work together on problem sets, so please don't make life unpleasant for all of us by breaking these rules. The penalties for cheating at Dartmouth are severe, starting with suspension and including expulsion. If you are unsure about anything, please ask.

Any hard-copy and electronic sources used must be properly cited and acknowledged. The only exceptions to this rule are as follows. It is not necessary to cite material which is copied from lecture notes or example code which we supply you in the course of this term.

Material copied or otherwise drawn by you personally from the course textbook need only be cited in general terms, e.g., "this code sample is from the textbook."

### *Previous offerings*

You may not consult any solutions from previous offerings of CS 10 or the former CS 5 and 8, from which some material is adapted.

---

## ***Disabilities***

Students with disabilities enrolled in this course and who may need disability-related classroom accommodations are encouraged to make

an appointment to see the instructor before the end of the second week of the term. All discussions will remain confidential, although the [Student Accessibility Services](#) office may be consulted to discuss appropriate implementation of any accommodation requested.

---

### ***Religious Observances***

Some students may wish to take part in religious observances that occur during this academic term. If you have a religious observance that conflicts with your participation in the course, please meet with me before the end of the second week of the term to discuss appropriate accommodations.