# HH/PSYC 6273 3.0 A
# Computer programming for experimental psychology
# Fall 2019

**Instructor:** Richard Murray rfm@yorku.ca

**Lectures:** Thursdays, 11:30 - 2:30
BSB 159 (Hebb undergraduate/teaching computer lab)

**Website:** github.com/rfmurray/psyc6273

**Textbook:** MATLAB: A practical introduction to programming and problem solving, fifth edition (Attaway, 2018)

**Evaluation:** six tests (10%), two problem sets (40%), term project (50%)

**Overview.** This graduate course covers computer programming methods that are useful in experimental psychology. The course assumes no previous programming experience, and brings students to the point where they are able to write useful programs to advance their own research. Classes are held in a computer laboratory, and each class consists of a lecture followed by programming practice on assigned problems. Topics include the MATLAB programming language, data files, curve fitting, Monte Carlo simulations, statistical tests, journal-quality data plots, 2D and 3D graphics, and interfacing to external devices.

## LECTURE SCHEDULE

| | topic | readings | tests etc. |
|---|---|---|---|
| 5-Sep | introduction | chapter 1 | |
| | vectors and matrices | chapter 2 | |
| 12-Sep | scripts and functions | chapter 3 | |
| | if-elseif-else | chapter 4 | |
| 19-Sep | loops | chapter 5 | test 1 |
| | files, etc. | chapter 6 | project proposal |
| 26-Sep | the psychtoolbox | | |
| | | | |
| 3-Oct | curve fitting | | test 2 |
| | | | |
| 10-Oct | image matrices, plots | chapter 11 | test 3 |
| | | | |
| 17-Oct | *fall reading week; no class* | | |
| | | | |
| 24-Oct | data structures | chapter 8 | |
| | bootstrapping | | problem set 1 |
| 31-Oct | bootstrapping | | test 4 |
| | strings | chapter 7 | |
| 7-Nov | statistical functions | | |
| | | | |
| 14-Nov | simulations | | test 5 |
| | | | |
| 21-Nov | MEX files, the GUI | | |
| | | | |
| 28-Nov | overflow, review | | test 6 |
| | | | problem set 2 |

The term project is due on Thursday, December 12.

**Guidelines on plagiarism.** An important part of learning how to program is discussing problems with other people, and reading other peoples' code. This makes it important to think about what constitutes plagiarism. Here are some guidelines. You can discuss assigned problems with others as much as you want, and read each others' code, but in the end you must do your own work. If you cut and paste someone else's code, you are plagiarizing. If you find yourself looking at someone else's code while writing your own, you are probably plagiarizing. If you memorize someone else's code and type it in without understanding how it works, you are plagiarizing. You should think of computer programming as problem solving, and it is important that you provide your own solutions to assigned problems. That said, discussions are an important part of solving difficult problems, and it is inevitable and acceptable that different peoples' solutions will end up being similar in some ways.