Welcome to CS116x
CS 116 LEC 002
Global Business and Digital Arts (GBDA) only

Audience
This course is a continuation of the Fall 2014 offering of CS115x. If you didn’t take that course, don’t take this one.

Course staff
Instructor: Craig S. Kaplan
Email: csk@uwaterloo.ca
Office hours: Wednesdays 11–12, DC 2110
Course staff

Coordinator: Ahmed HajYasien
Email: ahajyasien@uwaterloo.ca

Instructional Assistant: Alyssa Jamal
Email: amjamal@uwaterloo.ca
Office hours: Mondays, 1–2:30, MC 4065

Instructional Assistant: Chrissy Schreiner
Email: lcschrei@uwaterloo.ca

Instructional Assistant: Aaron Voelker
Email: arvoelke@uwaterloo.ca

Teaching assistants

Undergraduate advisors

Organization

Lectures: MW 2:30pm–3:50pm, MC 4059
Labs: F 2:30pm–3:50pm, MC 3003 & MC 3027

Roughly ten assignments, due Tuesdays at noon

Midterm: Thursday, 5 March, 7:00pm–8:50pm, MC 4045, MC 4061

Final exam: TBA
Marking

You must pass the weighted sum of the exams:
\[ \text{if}(0.33 \times M + 0.67 \times F < 50) \{ \text{fail}(); \} \]

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participation</td>
<td>5%</td>
</tr>
<tr>
<td>Labs</td>
<td>5%</td>
</tr>
<tr>
<td>Assignments</td>
<td>30%</td>
</tr>
<tr>
<td>Midterm</td>
<td>20%</td>
</tr>
<tr>
<td>Final</td>
<td>40%</td>
</tr>
</tbody>
</table>

Assignments

**No Late Submissions!**
Submitted online using LEARN
All work done individually (labs and assignments)

Resources

Course web page
www.student.cs.uwaterloo.ca/~cs116/x/

LEARN
learn.uwaterloo.ca/

Piazza
piazza.com/class#winter2015/cs116x

Processing
processing.org

NO CHEATING!
Technology in class

You don’t have to become a computer scientist.

But some art and design problems are best solved using code.

Code is an infinitely flexible medium.

Philosophy

Books

Reas and Fry

Shiffman

Reas and Fry

Glassner

imaginary-institute.com
Philosophy

Learn how programs work and how developers think.

xkcd.com/1425

Content

Assume you understand the basics of programming.

How can we make programming more useful?
How can we solve more problems?

Make use of more built-in functions and add-on libraries.

Content

1. Input/Output
2. User interfaces
3. Physics and animation
4. Geometric context
5. Procedural content
6. Advanced programming
7. Image processing
8. Text processing
9. Structured data processing