CSC148-Section:L0301
Week#1-Friday

Instructed by
AbdulAziz Al-Helali
a.alhelali@mail.utoronto.ca
Office hours: Wednesday 11-1, BA2230.

Slides adapted from Professor Danny Heap and Jacqueline Smith slides winter17
Outline

• Introduction and Course Logistics
• Objects
Course Information

• Course Webpage
  • http://www.cdf.utoronto.ca/~csc148h/winter/

• Information Sheet
  • http://www.cdf.utoronto.ca/~csc148h/winter/courseinfo.pdf
Do I need CSC108 for CSC148?

• You should be familiar with general programming concepts.
  • We expect that you know the concepts in CSC108 such as:
    • if statements, loops, function definitions and calls, lists, dictionaries, searching, sorting, classes, documentation style.

• **IF not sign up for ramp-up session THIS WEEKEND:**
  • [https://doodle.com/poll/2arm5xn44zxn7zda](https://doodle.com/poll/2arm5xn44zxn7zda)
  • Saturday, January 6, 10am - 4pm
  • Or Sunday, January 7, 11am - 5pm
  • If you took CSC108 and did well you will not need the ramp-up session.
    • As slides will be posted on the course webpage.
What's CSC148 about?

• **How to Understand and Write a solution for a real-world problem**
  • Problem in English—write a solution in Python

• **Abstract Data Types (ADTs) to represent and manipulate information**
  • You have used built-in data types such as: int, str, bool
  • Now, create your own datatypes to:
    • Hide info/algorithm
    • Share with other using public interfaces and documentation

• **Recursion:** Clever functions that call **themselves** in their definition

• **Exceptions:** how to deal with unexpected situations
  • Learn how to write/read/use exceptions in python

• **Design:** how to structure a program to be easily fixed and maintained by
  • Making excellent documentation
  • Well organized code

• **Efficiency:** know and control how much resources: time and space your code consumes
Textbook and Computing

• We’ll provide slides and links to readings online relevant to our weekly topics

• By virtue of registering in this course, you will have a Teaching Labs account
  • it is vitally important that you set it up so that you are able to log in.
  • Gives you access to computing resources both remotely and within the Bahen building
  • Allows you to submit course work.
# Marking scheme

<table>
<thead>
<tr>
<th>Work</th>
<th>Due</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 labs/exercises</td>
<td>every week except weeks 1, 5, 9 and 11</td>
<td>16%</td>
</tr>
<tr>
<td>two assignments</td>
<td>A1, January 30th, 10 p.m.</td>
<td>10%</td>
</tr>
<tr>
<td></td>
<td>A2, March 6th, 10 p.m.</td>
<td></td>
</tr>
<tr>
<td>two face-to-face demos</td>
<td>February 1–2, during lab</td>
<td>12%</td>
</tr>
<tr>
<td></td>
<td>March 8–9, during lab</td>
<td></td>
</tr>
<tr>
<td>two term tests</td>
<td>T1, February 7th, during lecture time</td>
<td>24%</td>
</tr>
<tr>
<td></td>
<td>T2, March 14th, during lecture time</td>
<td></td>
</tr>
<tr>
<td>Final exam</td>
<td>some time in April</td>
<td>38%</td>
</tr>
</tbody>
</table>
Marking scheme

• Designed to place a relatively **low weight (38%) on the final exam**

• We aim **to give higher weight** to your **better work**.
  • For example, the weights of your two assignments sum to 10%, so we’ll give the best one a weight of 6% and the worst a weight of 4%.
  • Read the course information sheet for more details.

• In addition to the other requirements, must you **achieve 40% of the marks on the final exam** in order to **pass** this course.
Re-marks

• It is very important to us that your work is evaluated correctly, and that occasionally includes re-marking work to see whether we agree with the original grading.

• How?
  • All re-mark requests must be submitted on MarkUs within 7 days of when the relevant work is handed back.
  • Use MarkUs to request re-marks.
  • We promise to consider the request before we submit grades at the end of the course, but we cannot guarantee earlier than that.
Lateness, sickness, natural disasters

• We discourage late work since we have to
  • arrange in advance for grading it
  • we want to be able to discuss solutions soon after the assignment is due, while it is still fresh in everyone’s minds.

• Late assignments will be penalized at the rate of 5% per hour

• If you have special circumstances:
  • please contact us immediately (usually before the work is due)
  • fill out the “Request for special consideration,”
  • provide all supporting documentation

• We will do our best to ensure that your evaluation is not harmed by events that are not your fault.
Independent work

• Be sure to **give** full and generous **credit to any person or book** (except course instructors and teaching assistants) you consult in solving assignments. If you take notes when you consult a source, quote that source in full.

• Do **NOT** look at similar work by other students, in written or electronic form, since looking can easily **turn into plagiarism**

• Do **NOT** show your own assignments **to other students**

• **You can discuss general ideas but**
  • Take **a couple of hours’ break** before writing it up.
Email, piazza

• **Use Piazza**
  • for **questions** and **answers** that don’t reveal the details of assignments. You may use piazza in “demo” mode if you do not wish them to have access to your email.

• **Use Email:**
  • If you have a question that can’t be raised on piazza and is suitable for email, **BE SURE TO INCLUDE**
    • in the **Subject:** "CSC148," and something about your question,.
    • in the **Signature:**
      • your **full name** your **student ID**(full name is not enough, some students have the same full name)
Objects

```python
>>> a = A()
>>> b = A()
>>> c = A()
>>> b.set_x(3)
>>> c.is_a
False
>>> b.is_a
False
```
Objects

• Here are some built-in objects to fool around with:
  • >>> w1 = "words"
  • >>> w2 = "swords"[1:]
  • >>> w1 is w2
    • False
  • >>> import turtle
  • >>> t = turtle.Turtle()
  • >>> t.pos()
    • (0.00,0.00)
  • >>> t.forward(100)
Objects

Why??

```python
>>> n1=1
>>> n2=1
>>> n1 is n2
True
>>> n1==n2
True
>>> n3=257
>>> n4=257
>>> n3==n4
True
>>> n3 is n4
False
>>> s1='word'
>>> s2='swords'[1:5]
>>> s1 is s2
False
>>> s1 == s2
True
```
Objects

\[ n_1 \rightarrow 1 \]
\[ n_2 \]
\[ n_3 \rightarrow 257 \]
\[ n_4 \rightarrow 257 \]

```
if integer > 256
  Python creates a new object
```