

# CPSC 340 Assignment Submission Instructions for 2021W2

## 1 The assignments

- The code and data for assignment **X** is in **aX.zip** on the course website, under **assignments**.
- Make sure to do the version of the assignment from the current semester.
- Modifications/updates/clarifications after the assignment is first put online will be marked in **red**. Any major updates will be announced on Piazza.

### 1.1 Groups

- You can work on your own or in a group of 2, **except for the first assignment** which must be done alone.
- You can have different groups on different assignments.
- If you work in a group, check how to submit as a group on Gradescope in Section 4. Make sure you do this!

### 1.2 Grading

- All homework assignments have the same weight.
- We try to provide grades within 10 days of the initial assignment submission deadline.
- If you think there is a problem with your grade, you have one week to raise a concern from the time that your grades were posted. After that, your grade is final. See full policies in the syllabus.
- To raise a concern, use the **regrade request** on Gradescope.

## 2 Submission format

- Put your name(s) and student number(s) on the first page, as a backup.
- Your submission needs to be organized sequentially using the (sub-)section numbers in the assignment.
- Figures AND relevant code need to be placed in the appropriate location in the document.
- If you use information from students outside your group or from other sources, cite this at the start of each question. **You will receive a mark of 0 for the assignment (and possibly further consequences) if you are found copying from other sources without citation.**
- Make sure your answers are easy to find, clearly written and easy for the TA to understand. TAs can reduce your mark if your answers aren't clear or difficult to understand (even if correct).
- To reduce the chance of this happening, you should prepare your report in L<sup>A</sup>T<sub>E</sub>X. To help you with this, we will put the L<sup>A</sup>T<sub>E</sub>X file used to generate the assignment PDF on the webpage. The file is, e.g., **a1.tex**. You can use the `\begin{answer}` environment, or the slightly-shorter `\ans{}` command, to write your answers (appearing in green). Here is an example of a legible answer format:

Figure 1: Short answer example

## 1.2 Matrix Algebra Rules

Assume that  $\{x, y, z\}$  are  $n \times 1$  column vectors and  $\{A, B, C\}$  are  $n \times n$  real-valued matrices,  $0$  is the zero matrix of appropriate size, and  $I$  is the identity matrix of appropriate size. **State whether each of the below is true in general** (you do not need to show your work).

1.  $x^T y = \sum_{i=1}^n x_i y_i$ .

Answer: 234567

2.  $x^T x = \|x\|^2$ .

Answer: 234567

3.  $x^T x = x x^T$ .

## 2.1 Including code

There are several options for including the code into your answer file. When you submit code, **please only submit the code you wrote**, not the provided “boilerplate” code that we wrote for you.

### 2.1.1 minted

The nicest option is the `minted` package (package documentation, Overleaf’s description). If you’re compiling your document on Overleaf, it should Just Work™ to add `\usepackage{minted}` to your preamble, then include something like

```
\begin{minted}{python}
import numpy as np

def sort_really_really_fast(ary):
    while not np.all(np.diff(ary) >= 0):
        np.random.shuffle(ary)
\end{minted}
```

which will end up looking like

```
import numpy as np

def sort_really_really_fast(ary):
    while not np.all(np.diff(ary) >= 0):
        np.random.shuffle(ary)
```

You can also use `\inputedminted[firstline=12,lastline=23]{python}{code/filename.py}` to pull code in directly from your code file – but double-check that you’re including all the relevant bits and not too much boilerplate if you do it this way.

To use `minted` locally, you’ll need to `pip install pygments` and run `latexmk/pdflatex` with the `-shell-escape` flag. Getting the paths right might be annoying if you’re using an IDE instead of the command line, so you might want to consider another option below. If you do this, you should probably **comment out the `\usepackage{minted}` in the preamble** (or else you’ll still get an error about requiring `-shell-escape`).

### 2.1.2 listings

If you're having trouble with `minted`, a less-nice but easier-to-get-working option is the `listings` package (Overleaf description). To use this, just put `\usepackage{listings}` in the preamble, then use

```
\begin{lstlisting}[language=Python]
import numpy as np

def sort_really_really_fast(ary):
    while not np.all(np.diff(ary) >= 0):
        np.random.shuffle(ary)
\end{lstlisting}
```

which will end up looking like

```
import numpy as np

def sort_really_really_fast(ary):
    while not np.all(np.diff(ary) >= 0):
        np.random.shuffle(ary)
```

It's fine, but not as pretty as `minted` was. If you want, you can add some colour to it; see Overleaf's example.

You can use `\lstinputlisting[language=Python, firstline=12, lastline=23]{filename.py}` to include a whole file (again, double-check you're including all your code and not too much boilerplate).

### 2.1.3 Last resort

If for some reason `listings` is giving you issues, it's okay to just include a screenshot with `includegraphics`. Please try to make one of the above options work, though – if we're not sure about something in your code from just looking at it, we really don't want to have to re-type it to run it.

Figure 2: An example of code included via screenshot.

```
import os
import sys
import random
import math
import numpy as np
import skimage.io
import matplotlib
import matplotlib.pyplot as plt

# Root directory of the project
ROOT_DIR = os.path.abspath("../")

# Import Mask RCNN
sys.path.append(ROOT_DIR) # To find local version of the library
from mrcnn import utils
import mrcnn.model as modellib
from mrcnn import visualize
# Import COCO config
sys.path.append(os.path.join(ROOT_DIR, "samples/coco/")) # To find local version
import coco

%matplotlib inline

# Directory to save logs and trained model
MODEL_DIR = os.path.join(ROOT_DIR, "logs")

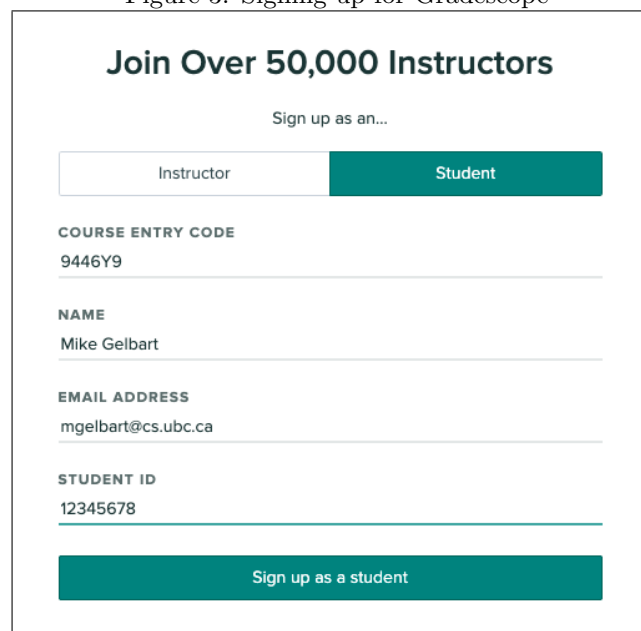
# Local path to trained weights file
COCO_MODEL_PATH = os.path.join(ROOT_DIR, "mask_rcnn_coco.h5")
# Download COCO trained weights from Releases if needed
if not os.path.exists(COCO_MODEL_PATH):
    utils.download_trained_weights(COCO_MODEL_PATH)

# Directory of images to run detection on
IMAGE_DIR = os.path.join(ROOT_DIR, "images")
```

### 3 Signing up on Gradescope

- The easiest way to set up for Gradescope is to just follow the link from Canvas.
- If that's not working for you for whatever reason, go to <https://www.gradescope.ca> to sign up. We are using the Canadian-hosted Gradescope; do **not** go to [gradescope.com](https://www.gradescope.com). The **course entry code** is **9DREZ9**.
- Your account on [gradescope.ca](https://www.gradescope.ca) is completely separate from any accounts you may have on [gradescope.com](https://www.gradescope.com).
- **Fill out your information as shown in Fig 3.**
- Use your UBC Student number in the **Student ID** field.
- If you already have an account on [gradescope.ca](https://www.gradescope.ca), make sure the information is up to date.

Figure 3: Signing up for Gradescope



The screenshot shows the Gradescope sign-up interface. At the top, it says "Join Over 50,000 Instructors". Below this is a "Sign up as an..." section with two buttons: "Instructor" and "Student". The "Student" button is highlighted in teal. Below the buttons are four input fields, each with a label and a value: "COURSE ENTRY CODE" with "9446Y9", "NAME" with "Mike Gelbart", "EMAIL ADDRESS" with "mgelbart@cs.ubc.ca", and "STUDENT ID" with "12345678". At the bottom is a large teal button labeled "Sign up as a student".

### 4 Submitting on Gradescope

- **Submit your assignments as a single PDF to Gradescope.** See Fig 4
- When submitting, you will be prompted to indicate which page contains which question. See Fig. 5
  - For each question, make sure to indicate **all** pages that correspond to that question.
  - Make sure all questions are assigned to at least one page.

**If you fail to do so, that may result in having a score of 0 for that question.**

- If you work in a group, **only hand in one assignment.** Use Gradescope's group feature. See Fig. 6  
<https://help.gradescope.com/article/m5qz2xsnjy-student-add-group-members>
- Assignments are due at 11:59pm Vancouver time on the submission deadline.

- You can submit each assignment multiple times. We recommend submitting a preliminary version early.

Figure 4: When uploading the file

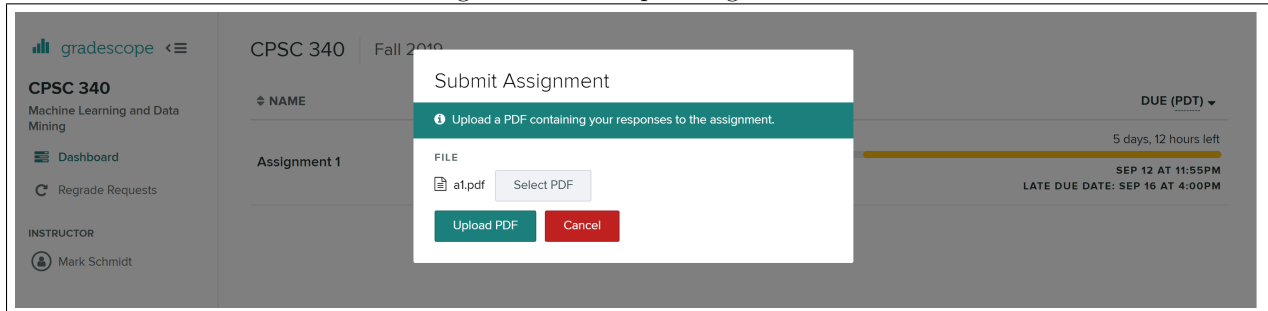


Figure 5: Assigning questions to pages

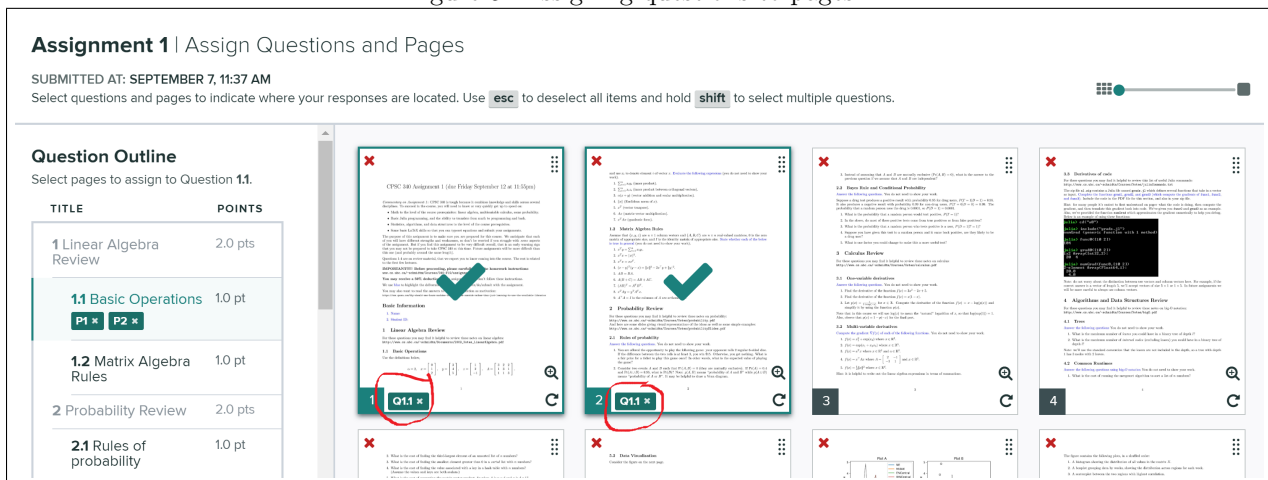


Figure 6: If you are working in pairs, *submit your assignment first* and then configure your group using the “Group Members” button. Select your partner using the dialog box. Make sure your partner has a valid email address.

