

# MCS 260 – Introduction to Computer Science – Fall 2020 – Emily Dumas

## Week 1 Worksheet

**Instructions** This worksheet is divided into two “tracks”. It is important to spend some of the discussion meeting working on each one. Follow your TA’s instructions on which to start with, and when to change.

It is OK if you are not able to complete the entire worksheet during the discussion meeting, but you *should* complete the rest of the worksheet during the week.

### Track 1: Software setup

If you haven’t already done so, work through the following MCS 260 startup tasks.

- (0) If you plan to use the ACCC Virtual Computer Lab, connect to it, log in, and complete all subsequent steps in the virtual environment it provides. However, it is recommended to use your own computer if possible.
- (1) Install Python 3.8 (or any version greater than or equal to 3.6).
  - Platform-specific instructions and install links can be found on the Getting Started Checklist on the [course web page](#).
  - If you are using the ACCC Virtual Computer Lab, Python 3 is already installed. Skip this step.
- (2) Open a terminal
  - In Windows, run Windows PowerShell (e.g. from the start menu)
  - In Mac OS X, run the application Terminal, which is located under Applications/Utilities.
  - In a graphical Linux desktop, a terminal application such as `gnome-terminal`, `xterm`, or `rxvt` is usually available from the application menu, or whatever system is used to launch applications.
  - In a Linux computer accessed through SSH, or without a graphical desktop, no action beyond logging in is necessary.
- (3) Run the Python 3 interpreter in the terminal by typing the interpreter command name followed by the Enter key. Be sure to check that Python 3 opens, and not Python 2. The version number appears in the first line printed after the interpreter command.
  - The possible commands for this are `python` and `python3`. Try both and see which open Python 3.
  - In Mac OS X, which ships with Python 2 pre-installed, it is likely that only `python3` will work.
  - **This step will be important again soon: Quiz 1 is going to ask you to prove that you have Python 3 working by opening it in a terminal.**
- (4) In the Python 3 interpreter, run the traditional first command

```
print("Hello world!")
```

by typing the text above followed by the Enter key. Then run the command `exit()` to quit back to the terminal.
- (5) Install a programming editor. If you don’t have one installed and don’t have a favorite, then install [Visual Studio Code](#).
  - If you are using the ACCC Virtual Computer Lab, then Visual Studio Code is already installed. Skip this step.
- (6) Open the editor (VS Code) and create a new file with the following contents:

```
print("Hello world!")
```

Save this file on the desktop as `hello.py`.

- (7) Switch back to the terminal, or open it again, as applicable. Change the current directory to the Desktop. In most newly-opened terminal applications on most platforms, one command that will achieve this is<sup>1</sup>

```
cd ~/Desktop
```

- (8) Run the script `hello.py` that you created earlier with the terminal command

```
python hello.py
```

or

```
python3 hello.py
```

depending on the interpreter command you determined earlier.

*Note 1:* This command must be run in the shell, and not in the Python interpreter. The Python interpreter will always show a prompt `>>>`, whereas a shell will usually show a prompt ending in `%`, `$`, or a *single* greater-than sign `>`.

*Note 2:* This simple script will run properly even if you accidentally use the Python 2 interpreter. In the future, however running scripts related to MCS 260 under Python 2 will create major problems. Thus it is important to make sure you are actually running Python 3. You can edit the script `hello.py` to add a check, as follows:

```
import sys
print(sys.version)
print("Hello world!")
```

This will print the Python version, followed by “Hello world!”. For example, on the Virtual Computer Lab systems, this script currently has output:

```
3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
Hello world!
```

## Track 2: Number systems

Work out these number conversion problems by hand. (This document doesn’t provide space to work; use a separate sheet of paper, a blank page in a tablet writing app, etc.) If you know how to check them with a calculator or with Python, you may do so afterward. The purpose of working them out by hand is to make sure you have a solid understanding of the theory of these number systems.

- (9) Convert `0b100000100` to decimal.
- (10) Convert `0x104` to decimal.
- (11) Convert the decimal number 66 to hexadecimal.
- (12) Convert `0xf0` to binary.
- (13) Convert the decimal number 20 to binary.
- (14) Convert `0x704` to decimal.
- (15) What is the decimal value of the largest 5-digit binary number?
- (16) What is the decimal value of the largest 3-digit hexadecimal number?

---

<sup>1</sup>See the slide presentation from the week 1 discussion meeting for more detail about `cd`.

**Revision history:**

- 2020-08-26 Revised directory change instructions to account for systems where the shell opens in a directory other than the home directory
- 2020-08-23 Initial release