15-110 Check1 - Written Portion

Name:

AndrewID:

Complete the following problems in the fillable PDF, or print out the PDF, write your answers by hand, and scan the results. When you are finished, upload your check1.pdf to **Check1 - Written** on Gradescope.

Written Problems

#1 - Writing Algorithms - 42pts

Can attempt after Algorithms and Abstraction lecture

In this problem, you will write plain-language algorithms (not code!) at three levels of abstraction. Assume all of your instructions will be provided verbally (no pictures). **Do not write more than 100 words per question** (and you can write much less!).

First, write an algorithm at a **low** level of abstraction that instructs a person on how to write the capital letter L. Assume the person you are instructing has almost no prior knowledge- they know directions (up/down/left/right), but nothing else about what pen and paper are, or how to write.

Second, write an algorithm at a **medium** level of abstraction that instructs a person on how to write the word 'ALL' in English, in all capital letters. This time you can assume the person you're instructing has a little more prior knowledge- what pen and paper are and how to use them, how to draw basic shapes, etc.- but you still should not assume they know how to write.

Finally, if you wanted to provide an algorithm on how to write the word 'ALL' in English at a **high** level of abstraction, what a**dditional starting knowledge** would you give the person being instructed?

#2 - Running Code - 28pts

Can attempt after Programming Basics lecture

The following question is intended to make you feel more comfortable with running code and encountering errors. In each of the following examples, copy the line of code into the interpreter (next to **>>>**) and press Enter to run it. Then copy the output in the interpreter into the space below the code and check a box below that to indicate whether the code ran successfully or raised an error.

```
5 / (4 - 2)
```

☐ Ran Successfully ☐ Raised an Error

```
"Hello World
```

☐ Ran Successfully ☐ Raised an Error

```
(8 + 3) < (5 * 2)
```

☐ Ran Successfully ☐ Raised an Error

```
8 + "two"
```

☐ Ran Successfully ☐ Raised an Error

#3 - Basic Programming Syntax - 30pts

Can attempt after Programming Basics lecture

Assume you've created some device to detect a medical condition. You've run a bunch of tests on study participants to determine how often the device detects the condition vs. not, and how often it gets the answer right vs. not. In other words, you've calculated the **true positive**, **false positive**, **false negative**, and **true negative** rates of this device. Read more here: https://en.wikipedia.org/wiki/Binary_classification

You've now set up four variables in your code that hold those four rates. You want to calculate the **recall** and **precision** of your device **using these variables** (not using numbers directly).

```
truePos = 15 # true positive - condition detected correctly
falseNeg = 8 # false negative - device did not detect condition that existed
falsePos = 2 # false positive - device detected a condition that didn't exist
trueNeg = 75 # true negative - 'no condition' detected correctly
```

Recall is calculated as $\frac{true\ positive}{true\ positive + false\ negative}$ (in other words, how many of the existing cases were detected?). Write a line of valid Python code here to calculate the recall of the device and store it in a variable **`recall`**.

Precision is calculated as $\frac{true\ positive}{true\ positive + false\ positive}$ (in other words, how accurate is the device when it gets a positive result?). Write a line of valid Python code here to calculate the precision of the device and store it in a variable **`precision`**.

Using these two variables, you can calculate the **F-score** of the device. This is a more general measure of the device's accuracy. The F-score is calculated as:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

Write a line of valid Python code to calculate and display the F-score of the device. This should use the two variables you defined and print `"Accuracy:"` followed by the result. For example, if the accuracy is 0.75, the code should print `"Accuracy: 0.75"`.