

15-110 Check6-1 - Written Portion

Name:

AndrewID:

#1 - Files in Code - 4pts

We've written some code to count the number of times the string "Hello World" occurs within a file:

```
f = open("sample.txt", "r")
print(f.count("Hello World"))
```

There is a bug in this code that will make it raise a runtime error, even if sample.txt is located in the same directory as the code. What is the bug, and how could you fix it?

#2 - Data Formats - 4pts

For each of the following data formats, identify whether it should be interpreted as a CSV, a JSON file, or as plaintext.

```
{ "restaurants": [  
  { "restaurant" : "Chipotle",  
    "date" : "11/01/19",  
    "menu" :  
      { "lunch" :  
        { "burrito" : 7.99,  
          "tacos" : 6.99,  
          "bowl" : 8.99 },  
        "dinner" :  
        { "burrito" : 8.99,  
          "tacos" : 7.99,  
          "bowl" : 9.99 } } },  
  { "restaurant" : "Sushi Fuku",  
    "date" : "11/01/19",  
    "menu":  
      { "lunch" :  
        { "bowl" : 8.99,  
          "sushi" : 10.99 },  
        "dinner":  
        { "bowl" : 10.99,  
          "sushi" : 12.99 } } } ]  
}
```

- CSV
- JSON
- Plaintext

City,Longitude,Latitude

Los Angeles,34°03' N,118°15' W

New York City,40°42' 46" N,74°00' 21" W

Paris,48°51' 24" N,2°21' 03" E

- CSV
- JSON
- Plaintext

#3 - Parsing Data - 7pts

You have been given a set of data about CMU classes in the following format (whitespace added for clarity):

| Professor, | ClassNum, | Days, | Time |
|--------------|-----------|--------|-------------|
| Reid-Miller, | 15106, | MWF, | 13:30-14:20 |
| Rivers, | 15110, | MWThF, | 14:30-15:20 |
| Reid-Miller, | 15110, | MWThF, | 15:30-16:20 |

Assume you've already split the string on "\n" and used the variable `row` to iterate through each class one line at a time.

How would you determine which department each class is in? Recall that the first two digits of the class number indicate the department.

- `row.split(',')[1][0:1]`
- `row.split(',')[1][0:2]`
- `row.split(',')[1][2:]`

How would you determine the start time and end time of a class and set those times as strings in the variables `start` and `end`? **Select all lines that are needed.** Assume that the code is run from the top selected line to the bottom selected line.

- `times = row.split(',')[0]`
- `times = row.split(',')[3]`

- `start = row.split('-')[0]`
- `start = times.find('-') - 1`
- `start = times.split('-')[0]`

- `end = times.find('-') + 1`
- `end = times.split('-')[1]`
- `end = start + "1:00"`

#4 - Components vs. Rules - 5pts

Let's say we want to design a simulation that determines how many students will sign up for a course during registration week. The simulation's time loop will loop over each sign-up time slot in order.

We need to design the model for this simulation. For each of the following values, would this value work better as a component of the model, or as a rule of the model?

Current length of the course's waitlist

- Component
- Rule

Students are more likely to sign up if a class is required for their major

- Component
- Rule

Number of students who are required to take this class, and haven't taken it yet, organized by sign-up timeslot

- Component
- Rule

Students are less likely to sign up for a class if the waitlist is long

- Component
- Rule

Information on whether or not the course will be offered again in the following semester

- Component
- Rule

#5 - Simulation Code - 10pts

We want to write code for a simulation that moves a circle from the left side of the screen to the right side of the screen in a 400px x 400px window. When the user clicks on the circle or presses 'Enter', the circle moves back to the left side of the screen.

For each part of the simulation (the Model, the View, the Time Loop, the Key Controller, and the Mouse Controller), select the line of code that needs to be included in that part.

Hint: if you're not sure, try implementing this using the simulation starter code!

Which line of code should be included in the **model**, in `makeModel(data)`?

- `x = 5`
- `data["left"] = 5`
- `canvas.create_oval(x - 20, y - 20, x + 20, y + 20)`

Which line of code should be included in the **view**, in `makeView(data, canvas)`?

- `data["left"] = data["left"] + 5`
- `canvas.create_oval(200 - 20, 200 - 20, 200 + 20, 200 + 20)`
- `canvas.create_oval(data["left"]-20, 200-20, data["left"]+20, 200+20)`

Which line of code should be included in the **time loop**, in `runRules(data, call)`?

- `data["left"] = 5`
- `data["left"] = data["left"] + 5`
- `x = data["left"] + 5`

How would you check if the user clicked in the circle in `mousePressed(data, event)`?

- `((data["left"] - data["x"])**2 + (200 - data["y"])**2)**0.5 <= 20`
- `((data["left"] - event.x)**2 + (200 - event.y)**2)**0.5 <= 20`
- `(data["x"]**2 + data["y"]**2)**0.5 <= 20`
- `(event.x**2 + event.y**2)**0.5 <= 20`

How would you check if the user pressed "Enter" in `keyPressed(data, event)`?

- `if (data["char"] == "Return"):`
- `if (data["keysym"] == "Return"):`
- `if (event.char == "Return"):`
- `if (event.keysym == "Return"):`

#6 - Identifying Features - 5pts

Say you wanted to make a machine learning algorithm that could identify the breed of a dog based on a set of features. List three important features you would include.

For each feature, specify whether that feature is **categorical**, **ordinal**, or **numerical**. You must have at least two different data types across your features.

Feature 1:

Feature 2:

Feature 3:

#7 - Machine Learning Categories - 10pts

For each of the following prompts, identify whether you should use **classification**, **regression**, or **clustering** to analyze the data and answer the question.

Using 15-110 students' three **numerical** exam grades to predict their end-of-semester **numerical** grades.

- Classification
- Regression
- Clustering

Using 15-110 students' three **numerical** exam grades to predict their end-of-semester **letter** grade.

- Classification
- Regression
- Clustering

Using 15-110 students' three **letter** exam grades to determine their end-of-semester **letter** grade.

- Classification
- Regression
- Clustering

Predicting Exam2's numerical grade using Exam1's numerical grade.

- Classification
- Regression
- Clustering

Identifying different (previously-unknown) categories of students in class based on their previous two homework grades.

- Classification
- Regression
- Clustering