PHY/AST/CS 104 FINAL

May 3, 2020

NAME: _____

Read these instructions:

1. **Before you begin please email me** and tell me you are starting.

2. You have **3 hours** to complete this exam.

3. There are various ways you can complete the exam. The most important criterion is that your answers should be legible and should be in numerical order. You may (a) print this out, write your answers on the sheet, and scan it at the end (scanning time does not count against your three hours.) (b) write your answers on a separate sheet of paper, and scan them (c) type your answers in a google doc or Word (or similar) and email them to you. For (b) and (c) make sure each question is very clearly labeled.

4. This is a closed book exam. **No notes or other aids are allowed**

5. The exam contains 26 questions numbered sequentially. Make sure you have all the pages. Each question's point value is next to its number. The total exam is worth **47 points**.

6. In order to be eligible for as much partial credit as possible, show all of your work for each problem, and **clearly indicate your answers**. Credit cannot be given for illegible answers.

7. Python code that you write should be as close to correct (runnable) as possible. Small syntax errors will not cost you points, but unclear indentation *will*.

8. Be sure to appropriately comment any code that you write. **You may lose points for correct but uncommented code.**

9. When finished please scan it and drop it in the dropbox I shared with you.

10. If you have a question please email me, but I may not respond in time. If you feel an exam question assumes something that is not written, write it down on your exam sheet. Barring some unforeseen error on the exam, however, you shouldn't need to do this at all, so be careful when making assumptions.

11. Be sure to read the entire exam before answering any questions, so you have time to think about the potentially tricky problems.

12. This examination is **NOT** to be shared with students who take this class in subsequent years, nor circulated in any manner. Thank you for abiding by the Honor Code of Haverford College in respecting this restriction.

13. Sign in ink below to indicate that you will **abide by the spirit of the Honor Code** in taking this examination:

1

Give the output of each set of Python commands.

Problem 1 (1 Points)   Python output

```python
def add10(x):
    x += 10
    return(x)
for s in range(5):
    print(add10(s))
```

Problem 2 (1 Points)   Python output

Careful with this one!!!

```python
x = [1,7,10,20, 37]
for y in x:
    print(x)
```

Problem 3 (1 Points)   Python output

```python
import numpy as np
x = np.array([2,7,8,11,27])
somenumbers = np.array([22, 32, 42, 52, 62])
i = np.where((x > 6) & (x < 15))
print(i)
print(somenumbers[i])
```

Problem 4 (1 Points)   Python output

```python
students = ["Evan", "Andrea", "Dan", "Shufan", "Tuck"]
for student in students:
    if "an" in student:
        print(student)
```

Problem 5 (1 Points)   Python output

There's a dataframe called "df" that I already read in that looks like this:

```
   Name        Age     City
0     Alexa    10      Chicago
1     Bob      12      Rio
2     Clarke   13      Tokyo
```

2

What's the output of this code? Note that on the last page of the exam I gave you some of the output of a web search for panda data frame loc, in case that helps you.

```
df.loc[df['Age']<13]
```

Problem 6 (1 Points)   Python output

This one is tricky! There's a twiddle before one of the conditions.

```
df.loc[~(df['Name']=='Alexa')&(df['Age']<13)]
```

Problem 7 (1 Points)   Python output

```
df.iloc[:,0:2]
```

The next 3 questions refer to the following. Assume the code below has been executed.

```
# Import pandas library
import pandas as pd

# initialize list of lists
data = [['tom', 10], ['nick', 15], ['juli', 14]]

# Create the pandas DataFrame
df = pd.DataFrame(data, columns = ['Name', 'Age'])
```

Problem 8 (1 Points)   Numpy array manipulation

What is the output of the following code:

```
ages = np.array(df.Age)
print(np.where(ages<15))
```

Problem 9 (1 Points)   Data frame manipulation

What is the output of the following code:

```
print(df[df.loc[:,'Age']<15])
```

Problem 10 (1 Points)   Data frame manipulation

What is the output of the following code:

```
print(df[df.iloc[:,1]<15])
```

Problem 11 (1 Points)   Problem code

This code will produce an error. Edit the code so that it doesn't.

```
def myfunc(x,y):
    z = x*y
    return(z)
print(myfunc(x,y))
```

Problem 12 (2 Points)   Code that works but probably shouldn't

Edit the code so that the function works in general, not just for the code below.

```
import matplotlib.pyplot as plt
x = np.arange(100)
y = x*x
"""This function makes a plot of any two arrays you give it.  The first
one is the x-axis and the second one is the y-axis."""
def myfunc(a,b):
    plt.plot(x,y, '.')

myfunc(x,y)
```

Problem 13 (2 Points)   Recursion

What's the output of this code:

```
def fact(n):
    if n == 1:
        return 1
    else:
        return n*fact(n-1)

fact(4)
```

Problem 14 (3 Points)   Redshift Slicing the Universe

Why would restricting redshift to a number between .3 and .4 (for example) give you a "slice" of the universe?

Problem 15 (6 Points)   Reading CSVs

Write a function that takes a filename and two strings as input, and reads in the CSV file found at that filename. It takes the two columns named by the two strings, and multiplies them together, element by element. The function returns an array that is the product of those two columns.

To be clear about what I mean by multiplying them together, if one column is [1,2,3] and the other column is [2,3,4] then the output of the function would be [2, 6, 12].

Problem 16 (1 Points)   Pathological ways to call functions

Imagine this function is defined in one cell of your notebook. What does the function do? (The next 3 questions also refer to this same function.)

```
def myfunc(x,y):
    z = x+y
    return(z)
```

In each of the 3 subsequent cells (which use the function above) say what the output of each cell will be? (At least one of them will give you an error.)

Problem 17 (1 Points)

```
print(myfunc(3,2))
```

Problem 18 (1 Points)

```
print(myfunc(x,y))
```

Problem 19 (1 Points)

```
a=7
b=1
myfunc(a,b)
print(z)
```

Problem 20 (4 Points)   Dataframe manipulation

Note that on the last page of the exam I gave you some of the output of a web search for panda data frame loc, in case that helps you.

I created a dataframe below. The question asks you to modify it. Please don't modify my code below, but rather create your own code to modify my dataframe.

```
>>> df = pd.DataFrame({'My First Column': [4, 5, 6, 7],
                       'Another Column': [10, 20, 30, 40],
                       'The Last Column': [100, 50, -30, -50]})

>>> df
Out[2]:
   My First Column  Another Column The Last Column
0     4                        10                      100
1     5                        20                       50
2     6                        30                      -30
3     7                        40                      -50
```

Create a new datatable called "mytable" that only contained the columns "My First Column" and "The Last Column" and calls them "first" and "last" instead. There are mutiple ways of accomplishing this. You only need to write one. On the last sheet of the exam there's some output from the web that you might find useful.

The next four questions refer to a class called "Rocket" defined below.

```
class Rocket():
    # Rocket simulates a rocket ship for a game,
    #  or a physics simulation.

    def __init__(self, x, y):
        # Each rocket has an (x,y) position.
        self.x = x
        self.y = y

    def move_up(self):
        # Increment the y-position of the rocket.
        self.y += 1
```

Problem 21 (3 Points)   Add a method

Add a method to the class called "move_down". You can add it in the space above.

Problem 22 (2 Points)   Create two objects

Create two objects in the class Rocket. One that starts at position (3,1) and one that starts in position (4,2).

Problem 23 (3 Points)   Using methods and employing loops

Move the first rocket up three times. Move the second rocket down three times. (You're good at loops - do this in a loop!)

Problem 24 (3 Points)   Write a method

Write a method that calculates the distance between two rockets. I'll give you a hint of how to start (don't forget this line would be inside the class):

```
    def get_distance(self, other_rocket):
```

Also here's a reminder of how to calculate distance between two coordinates (x_1, y_1) and (x_2, y_2).

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

8

Problem 25 (1 Points)   Call your new method

Even if you didn't write the get_distance method, write down how you'd call it to get the distance between the two rockets.

Problem 26 (3 Points)   Numerical Integration

Write code that calculates the area underneath the curve that is plotted here:

   x = np.arange(100) y = np.exp(-x) plt.plot(x,y)

Here's some of the web page on panda data frame "loc" in case it helps you.

```
pandas.DataFrame.loc
Access a group of rows and columns by label(s) or a boolean array.

.loc[] is primarily label based, but may also be used with a boolean array.

Allowed inputs are:
```

- A single label, e.g. 5 or 'a', (note that 5 is interpreted as a label of the index, and never as an integer position along the index).
- A list or array of labels, e.g. ['a', 'b', 'c'].
- A slice object with labels, e.g. 'a':'f'.
- Note that contrary to usual python slices, both the start and the stop are included
- A boolean array of the same length as the axis being sliced, e.g. [True, False, True].
- A callable function with one argument (the calling Series or DataFrame) and that returns valid output for indexing (one of the above)

Examples Getting values

```
>>>df = pd.DataFrame([[1, 2], [4, 5], [7, 8]],
      index=['cobra', 'viper', 'sidewinder'],
      columns=['max_speed', 'shield'])
df
            max_speed   shield
cobra               1        2
viper               4        5
sidewinder          7        8
```

Single label. Note this returns the row as a Series.

```
>>>df.loc['viper']
max_speed    4
shield       5
Name: viper, dtype: int64
```

List of labels. Note using [[]] returns a DataFrame.

```
>>>df.loc[['viper', 'sidewinder']]
            max_speed   shield
viper               4        5
sidewinder          7        8
```