PHOT 110: Introduction to programming

Final exam example questions

Michaël Barbier, Spring semester (2023-2024)

Before you start

The final exam counts for 50% of your total grade of the course. You will get 3 hours to complete your exam. The exam is performed on the computer, and you do not need to provide any answers on paper. There is one question (the third question) which asks to correct a script which contains various errors, other questions require you to write Python scripts.

During our final exam, there will be a very short (about 5 minutes) oral part. This part is only about the projects, you will be asked a couple of short questions about the project: the code you wrote or the accompanying report. If required you can prepare the answer on paper, you also will have a print out version of your report. You do not need to prepare for this oral part, the questions' only purpose is to see whether you understood what you did in the project.

Take the following points into account before you start the exam.

- For those who use their own computer/laptop: you do not need internet, we will ask you to switch of your WiFi before the exam starts. The exam files are distributed via USB drive. Copy all files from the USB drive to your computer before you start:
  - The questions: `Final_exam_questions.pdf`
  - The script with errors for question 3: `script_with_errors.py`
  - Three cheat-sheets: for Python, Numpy, and Matplotlib.
- For those who work on the desktop computers in the lab: all the required files should be found on the Desktop.
- Make sure that the `Numpy` and `Matplotlib` libraries are installed already (we will test this together before the exam starts).
- At the end of the exam we will go around with USB drives to collect the exams: put your solutions (and output files) in a folder which has both your name and your student number in the folder name, for example "Michael_Barbier_30029034".
- Let us know if during the exam there is any issue with the computer, PyCharm, or libraries. We will try to verify this up front, but please inform us in case of any issues.

You will be asked to save plots to your folder, this can be done using the `savefig` method, see the following example (the output of this example is a simple line plot):

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()
x = [1, 4, 3, 0]
y = [2, 2, 5, 4]
ax.plot(x, y)
fig.savefig("output_plot_example.png")
```

Questions

Please solve all of the following 7 questions. Each question is worth an equal amount of points (out of 100).

Question 1:

Write a script that prints all the strings in a list of strings after appending them to the string: `"King Arthur is "`. Use a loop (e.g. `for` or `while`) structure to automate the process. Your solution script should print each sentence on a separate line. If you use the following list of strings:

```
words = ["the greatest", "a mythological figure", "some old knight"]
```

then the output should look similar to:

```
King Arthur is the greatest.
King Arthur is a mythological figure.
King Arthur is some old knight.
```

Save your solution as a script with file name: `solution_1.py`.

Question 2: Cookie vending machine

Make a script that prompts a user repeatedly to input how many cookies he/she wants until the vending machine has no cookies left. For this, prompt the user to give a number between 1 and the number of cookies left (which you calculate). Tell the user each time how many cookies are left. If the number provided by the user is not a valid number (too large, too small, or not a number), tell the user that the input is invalid and prompt the user again. Stop the program when all cookies are finished. This is example output of a correct script (with a starting value of `N = 12` cookies):

```
How many cookies do you want to buy (12 left): 5
Here are your cookies! Have a good day.

How many cookies do you want to buy (7 left): 10
Sorry, your request is invalid. Please fill in a valid amount.

How many cookies do you want to buy (7 left): five
Sorry, your request is invalid. Please fill in a valid amount.

How many cookies do you want to buy (7 left): 7
Here are your cookies! Have a good day.

Unfortunately we are out of cookies!
```

Save your solution as a script with file name: `solution_2.py`.

Question 3: Correct a Python script

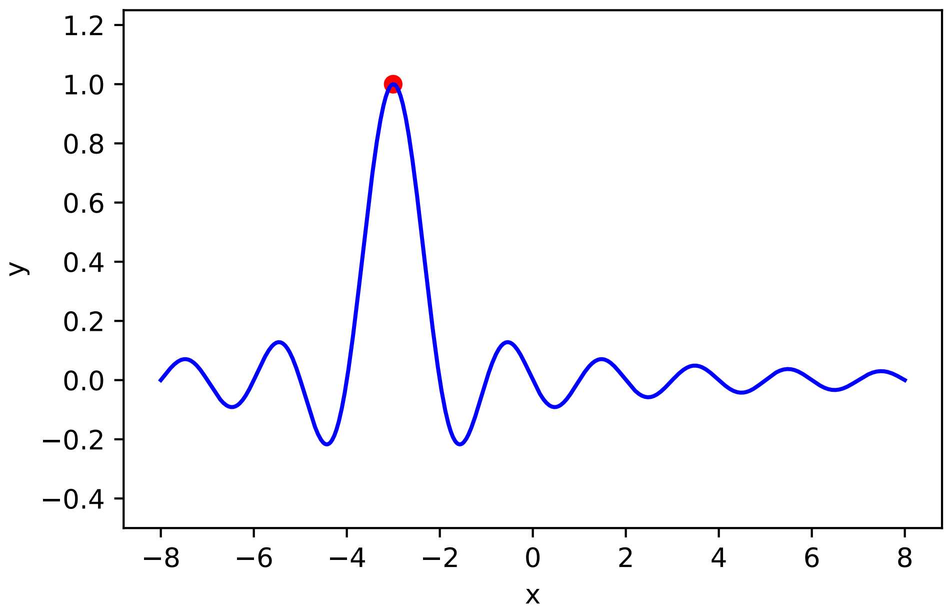
Open the script with name: `script_with_errors.py` and correct the errors. Save the corrected script with file name `solution_3.py`.

The corrected script should plot a sinc curve with equation:

$$y(x) = \text{sinc}(x + 4) = \frac{\sin(x + 4)}{x + 4}$$

Whereby it uses the Numpy `sinc` function. This graph is then saved as a png-file with file name `output_script_with_errors.png` to the hard disk.

The output graph should look as the plot below:



Question 4: 2D density plot

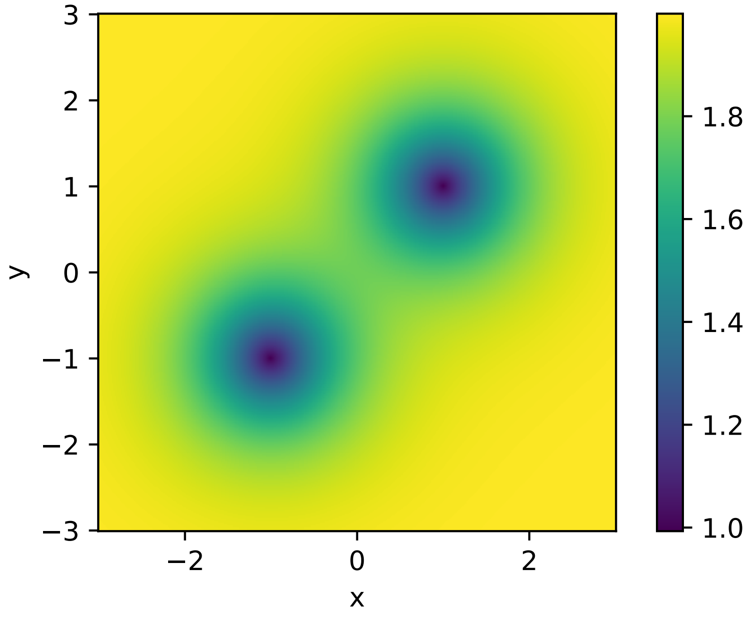
Plot $z(x, y)$, the superposition of two functions (forming a double well) $z(x, y) = z_1(x, y) + z_2(x, y)$ with equations:

$$\begin{cases} z_1(x, y) = \tanh(r_1) = \tanh\left(\sqrt{(x-1)^2 + (y-1)^2}\right) \\ z_2(x, y) = \tanh(r_2) = \tanh\left(\sqrt{(x+1)^2 + (y+1)^2}\right) \end{cases}$$

whereby you can use the tangent hyperbolic function of Numpy: `tanh()`. Plot the resulting function $z(x, y)$ as a density plot. Remember that you can make a density plot (with a colorbar) using the commands:

```
fig, ax = plt.subplots()
pc = ax.pcolormesh(xx, yy, zz)
fig.colorbar(pc, ax=ax)
```

For each of the functions you can use the same (x, y)-values within an 2D interval/domain with $x \in [-3, 3]$ and $y \in [-3, 3]$ (make your take a sufficiently high number of x values to obtain a smooth density plot). **Save the plot** under the file name: `output_plot_double_well.png`. Save your solution as a script with file name: `solution_4.py`. The output of the script should look as in the plot below:



Question 5: Creating and using modules

Create a **module** (a separate script file) with file name `module_area.py` which helps to calculate the areas of a square and a circle. The module should contain three functions:

- `area_square(side_length)` which accepts a single floating point number representing the side length of the square. The function should return the area.
- `area_circle(radius)` which accepts a single floating point number for the radius of the circle. The function should return the area of the circle. Remember that the area A of a circle is given by $A = \pi r^2$

Afterwards, import and use this module in a script (with file name: `solution_5.py`) that prompts a user to give the radius of a circle, then uses the above module to calculate its area, and finally prints the area of the circle. See the following example output of a correct script:

```
Please give the radius of the circle: 4.5
The area of a circle with radius 4.5 is: 63.61725123519331
```

Question 6: Dictionaries

Consider the following Python list where each item is a dictionary containing some statistical information about three football players: their name, the number of games they played, and the total amount of goals scored by them. Add this list to your script.

```python
player_list = [
  {"name": "Mehmet", "matches": 4, "goals": 12},
  {"name": "Ali", "matches": 5, "goals": 23},
  {"name": "Meryem", "matches": 3, "goals": 18}
]
```

Then make the script print out all the player names with their average scored goals per game (do this in an automated manner, by using a loop to iterate over all players in the list). To compute the average you divide their number of goals by the games they played. The output of a correct script should be the following (for the given player information):

```
Mehmet has an average of 3.0 goals/game
Ali has an average of 4.6 goals/game
Meryem has an average of 6.0 goals/game
```

Save your script under the name `solution_6.py`.

Question 7: Classes

Create a class named `Hotel` which has four attributes

- `hotel_name`: the name of the hotel,
- `n_rooms`: the number of guest rooms (all single rooms),
- `n_guests`: the current number of guests (an integer between 0 and the number of rooms inclusive).
- `guests`: a list containing the names of the guests.

where the first guest gets room number 1, the second guest gets room number 2, etc.

Give the class a constructor that accepts two arguments (in addition to the mandatory `self` argument): the hotel name and the number of rooms of the hotel. Then add two methods:

- `book_room(self, name)`: which books a room under the guest's name.
- `print_rooms(self)`: which prints an overview of the rooms and their guests.

Then use the class to make two hotel objects, one for a hotel at the town-square called "Plaza Hotel" and one for the hotel in the nearby forest called "Forest Hotel". The former hotel has 5 rooms and the latter only 2 rooms. Then try to book guests in the two hotels. Do the above using the following code:

```python
# Add here your class definition
#  ...

# Create the Hotel objects
hotel_plaza = Hotel("Plaza Hotel", 5)
hotel_forest = Hotel("Forest Hotel", 2)

# Bookings at Plaza Hotel
hotel_plaza.book_room("Yeliz")
hotel_plaza.book_room("Shana")
# Print the summary of Plaza Hotel
hotel_plaza.print_rooms()

# Bookings at Hotel Forest
hotel_forest.book_room("Rick")
hotel_forest.book_room("Mortimer")
hotel_forest.book_room("Bob")
```

For a correct script this should give you the following output:

```
Welcome at the Plaza Hotel Yeliz, your room number is 1
Welcome at the Plaza Hotel Shana, your room number is 2
--------------------------------------------------------------------------------
SUMMARY OF PLAZA HOTEL
Rooms booked: 2 / 5
The current hotel guests are:
Room 1: Yeliz
Room 2: Shana
--------------------------------------------------------------------------------

Welcome at the Forest Hotel Rick, your room number is 1
Welcome at the Forest Hotel Mortimer, your room number is 2
Sorry Bob, there is no available room at the moment.
```

Save your script under the name `solution_7.py`.