Programming Workshop: API and Project 1

Intermediate Workshop #5
17 October 2016

Outline

1. Speed Exercises
2. Solving problems using programming:
    a. Is problem suitable to approach with programming?
    b. Steps to approach problem with programming
    c. Example approach
3. Project 1 (if time)

Speed Exercises

1:

Make a function that accepts a list of strings and sorts them in alphabetical order.

300 seconds

Problem Solving with Programming: Criteria

1. Can the problem be numerically solved? (EX, “what is the meaning of life” can’t)
2. Can you output the answer meaningfully? (EX, “42” is meaningless)
3. Is a programmatic answer acceptable to your audience?
4. Do you understand the data you’re given?
5. Do you have the data needed? If not, can it be assumed, found, or can missing components be ignored?
6. Will the job complete in a reasonable time?
7. ***Has someone already solved this problem?***

Problem Solving: Global Steps

1. Understand what “solved” means for the problem
2. Understand what data is provided
3. Explore and understand the format of the data
4. Plan algorithm or technique (what’s needed to get from data to solution?) -- more on next page
5. Write code
6. Debug
7. Test code for validity
8. Get solution to the problem or start process over with modified understanding

Problem Solving: Planning the Algorithm

1. Have a precise understanding of:
    a. What's needed to solve the problem (format **and** content)
    b. Data format
2. If needed, translate data into program-useable format (often challenging)
3. Use **numeric** techniques to translate from data to solution
4. Present user with solution **in a meaningful way**

Problem Solving: Planning the Algorithm (cont'd)

Example:

Take start temperature, material, and approximate shape of sample. Determine temperature distribution over sample if one end is placed in bath of a certain temperature. Graph it in 3D. Update the graph once per second per time quantum.

Initial data is found in a file in the described order, one value per line. Temperature is in Kelvin, the material is the chemical symbol(s), and the approximate shape is a string.

Problem Solving: Planning the Algorithm (cont'd)

1. Have a precise understanding of:
    a. What's needed to solve the problem (format **and** content)
    b. Data format

Example:

What's needed: temperature distribution (graphed)

What's given: formatted file

Problem Solving: Planning the Algorithm (cont'd)

2. If needed, translate data into program-useable format

Example:

Given file of strings.

- Translate temperature to float, understand it's in Kelvin, ensure it's in expected ranges
- Translate from the chemical name to the desired chemical properties. Consult database.
- Translate from the approximate shape to the appropriate spatial distribution components (IE, iterate over bar differently than over sphere)

Problem Solving: Planning the Algorithm

3. Use **numeric** techniques to translate from data to solution

Example:

Using an iterative algorithm for temperature distribution given time, space, and conditions, find temperature distribution.

Problem Solving: Planning the Algorithm

4. Present user with solution **in a meaningful way**

Example:

Problem asks us to graph. Do so.

Project 1

In groups of 2 or 3:

- Using data in x_y_5773672638_duplicates_NO_ERROR_FOUND.txt and the corresponding description, collect both sets of points.
- For the x coordinates, find the median and the mode.
- For the y coordinates, find the mean and the range.
- For each pair of coordinates, find the biggest difference out of all data.
- Print all of these values to the console, and be sure to say what you're printing!
    - IE, do:

      X coordinate median = #
      X coordinate mode = #
      Y coordinate mean = #
      Y coordinate range = from # to #

    - do not:

      #
      #
      #
      from # to #

Next Week:

- Finish Project 1
- How to approach programming problem