EECS 168/169 SI Session

Cody Ptacek :)

Activity

Background

- One thing that I love about programming is that there is very often more than one way to solve the same problem.

Task

- Starting on your own, try to solve the board work on the right.
- After 5-10 min (or when everyone is done), turn to a partner & compare code.
    - What things did you each do differently?
    - Can you think of more ways to solve the problem?

Boardwork:

- Prompt the user for their name.
- Prompt the user for their age in years.
- Print the user's name followed by their approximate age in seconds.
- Hints:
    - 365.25 days in a year
    - 24 hours in a day
    - 60 minutes in an hour
    - 60 seconds in a minute

Example output:

```
Enter your name: Cody
Enter your age: 26
Hello, Cody, you are 819936000secs old!
```

Activity

Things that could be done differently:

- Different variable names
- Storing age in seconds as a variable
- Storing the conversion formula as a variable
- Concatenating strings with ‘+’ instead of using print’s ‘,’
- Storing age as a float vs an int.

Discussion

- Go through each of the points listed on the right. What are strengths and weaknesses of each different approaches?
    - Are there contexts where one approach might be better or worse than other approaches?

Activity: Writing While Loops

Background

- Just a boardwork type exercise. What else can I say besides that it’s just good practice.
- ¯_(ツ)_/¯

Task

- Using a while loop, print out a triangle made of ‘&’s. This triangle will start out with one ‘&’ in the top row and progress to have eight ‘&’s in the bottom row.

Example output

-
  ```
  &
  &&
  &&&
  &&&&
  &&&&&
  &&&&&&
  &&&&&&&
  &&&&&&&&
  ```

Activity: Code Correction

Background

- While writing code for labs, it is very likely that you will run into errors. It is important that you practice reading code so that you can find and fix these errors.
- Later on in life, when you're working on larger projects in groups, you will need to be able to read and help fix other people's code.

Task:

- Fix the mistakes in the code on the next slide. There are 9 mistakes.
- Some mistakes will cause errors. Others are just violations of good coding practice.
- See if you can find all 6.

Code

```
x = int(input("what is the speed of light in m/s?: "))

k = 0

while k<10:
if k = 0:
   print("That has no energy!")
if k != 0:
   l = k * x**2
   print("A"+k+"kg object would have energy of"+l+"newtons!")
```

Activity: Which Loop is Best

- The following few slides will present several different scenarios that will require loops to solve.
- You will not be tasked to solve them. Instead, you will be asked to explain which type of loop (for_in vs while) would be best for that specific task.
    - Some tasks can be completed with either looping tool just as easily. For those ones, either will work.
- Be prepared to justify your answer.

A loop that iterates through a string provided by the user and counts how many occurrences of the letter ‘a’ are in the string?

A loop that prints all numbers from 1 to 1000 that are divisible by 7.

A loop that repeatedly prompts the user to donate money to the Order of the Phoenix until the user inputs a number greater than 200 Galleons.

A loop that iterates through a string recording the position of the last occurrence of the character ‘?’ within the string.

A loop that prompts the user with a menu until the user decides to exit the menu.

A loop that prompts the user to list their top five movies or books, asking for each one at a time. For example:

- What is your #1 favorite book/movie/tv show?:
  What is your #2 favorite book/movie/tv show?:
  What is your #3 favorite book/movie/tv show?:
  What is your #4 favorite book/movie/tv show?:
  What is your #5 favorite book/movie/tv show?:

Activity: Boardwork/Get to Know Each Other

Background

- I actually like the last one as a board work.
- Complete a board work with the functionality described to the right. (It's similar to the last question)
- Once you finish, go through your neighbor's code answering the questions & outputting the final string as they have written it.

Board Work

A loop that prompts the user to list their top five movies or books, asking for each one at a time. For example:

- What is your #1 favorite book/movie/tv show?:
  What is your #2 favorite book/movie/tv show?:
- etc.

Store the answers in one string concatenating them in the following style.

"1) answer1; 2) answer2; etc.

Print this output.

Activity: Number Guesser

Task

- Construct a program that prompts the user to guess the number of jelly beans in the jar.
- The loop will continue until the user guesses the correct number of jelly beans.
- The loop will tell the user if their guess was above or below the correct number.
- If the user guesses a number within 10 beans of the correct answer, the program will additionally say, “So close! You’re almost there!”
- Congratulate the user when they finish and tell the user how many guesses it took.

Example Output:

```
Guess how many jelly beans in the jar: 99
Too few! Guess again!
Guess how many jelly beans in the jar: 1000
Too many! Guess again!
Guess how many jelly beans in the jar: 591
So close! You’re almost there!
To few! Guess again!
Guess how many jelly beans in the jar: 593
Correct! It took you 4 guesses.
```

Activity: File IO with words!

Task

- Create a program that opens a txt file provided by the user that contains words separated by new lines.
- Store these words in a list
- Calculate the following stats.
  - The number of occurrences of the word ‘the’ case insensitively.
  - The number of occurrences of the word ‘taco’ case insensitively.
  - If ‘And’, ‘But’, or ‘Or’ appear in the list, case sensitive, tell the user they shouldn’t start sentences with conjunctions.

Example input_file.txt

```
I
ate
the
taco
Bill
made
for
Suzie.
But
I
am
weird!
```

Program Output

```
Input a file name: input_file.txt

The word ‘the’ occurred 1 times!

The word ‘taco’’ occurred 1 times!

Don’t start sentences with conjunctions!
```

Final Activity: Write a Test Question!

Background

- Tests! They're on the horizon! It might seem like you have plenty of time, but it's good to start studying now.
- Gibbon's test will involve questions similar to what I've shown you today, asking you to interpret code, write code, or answer conceptual questions about code.
- It's a good skill to learn how to think about what test questions might look like and what skills/knowledge is required to answer them.

Task

- Write a test question! Be sure to include the/an answer - but not right next to the question.
- Examples
  - board work
  - Interpret
  - Multiple choice
  - etc.
- What skills/knowledge are required to answer your question.
- Turn to a partner and present them with your question.