YEAH! Hours Assignment 2

Elina Thadhani

Part 1: Sandcastles (Tracing)

Tips and Tricks for Tracing:

- Write it out by hand!
- Keep track of the value of your variables
- Keep a running “console” of what you would have printed on your screen!

Console Programming Quick Guide

Some key functions you can use:

answer = **input(“Enter prompt here:”)** # this function gets an input from users

When you get an answer from the user, the variable will be a string or word. So remember if you are looking for a number to convert to a **float** or an **int**. Conversion looks like this:

x = int(input(“Enter your number here: ”))

x = float(input(“Enter your number here: ”))

Console Programming Quick Guide (cont)

Printing to the console requires a string! If you have a number print it in this manner:

```
X = 3

print(“the number is”, x )  #here you print two types

print(“the number is :” + str(x) + “.”)        #here you concatenate to add punc.
```

Part 1: Sandcastles (subtract_numbers.py)

Goal: to take in two inputs from the user (numbers) and then print the difference of the two numbers to the console.

- Check out lecture 5 if you are confused here there is a really similar problem with getting the sum of two numbers!

Part 1: Sandcastles (liftoff.py)

Goal: to print out in **descending** order the numbers from 10 to 1 and then write “Liftoff!”

This will include a for loop using **range**. Think about here how you can use the fact that in a for loop your variable changes by 1 each iteration of the loop.

```
for i in range (10):
    print(str(i))
```

Would print out:

1 2 3 4 5 6 7 8 9 10

Part 1: Sandcastles (random_numbers.py)

Goal: print 10 random integers between 0 and 100 inclusive. Use a constant NUM_RANDOM which in the example = 10, and constants MIN_RANDOM and MAX_RANDOM to determine the range.

X = random.randint(MIN,MAX)

will set x equal to a random number between MIN and MAX inclusive. Now think about how you could repeat this 10 times (or NUM_RANDOM number of times)?

Part 1: Sandcastles (moon_weight.py)

Goal: Given a user input of their weight, print out their weight on the moon.

Remember to :

- Use constants here!
- Notify the user when their weight isn't a valid weight!

Part 1: Sandcastles (pythagorean.py)

Goal: Given two user inputs of a and b as floats, give the solution of c.

Remember to convert the input to a float here using the technique we talked about in console overview!

Part 2: Khan-sole Academy

Goal: Generate simple addition problems for a user, reads in an answer and checks if the user is right or wrong. Give the users questions *until* the user has gotten 3 problems correct in a row!

We know from the sandcastles how to:

a) Generate random numbers
b) Get user input
c) Check if the user input is “valid” (in this case equal to the right answer)

All we need to do is repeat these steps until the user has gotten 3 valid answers in a row

Part 2: Computing Interest

Goal: Help a user compute the interest in their bank account over time

1. Ask the user for an initial account balance
2. Ask the user for a starting year and month
3. Ask the user for a end year and month
4. Ask the user for an interest rate
5. Print out monthly balance for each month in this range **including** the end year/month
6. Repeat 4-5 until the user enters an interest rate of 0%

Part 2: Computing Interest

Edge cases in collecting the inputs:

BEFORE you actually do the math, make sure the inputs are valid:

1) Is the start year/month before the start end/month?

When does the program end???

Part 2: Computing Interest

Skeleton:

1) Get the time and account values from the user (make sure the dates are valid)
2) Get the interest rate
3) Compute and then print the account value for each month in the range
4) Repeat steps 2 and 3 until interest rate = 0

Part 2: Computing Interest

Computing the interest rate through the months:

Note that the month will go up from 1-12 and then when the month = 12, in the next step you should “reset” the month counter to be 1 and the year to be +1 to simulate a new year!

Style Tips and Tricks

For Khan-sole Academy no need to decompose!

For Computing Interest, decomposing may help especially for printing!

USE CONSTANTS! Constants are great in making your code more readable!

Remember that to print a number you have to convert to a string

Good Luck!