Lecture 13:

Functions II

CPSC230 Computer Science I

Dr. Trudi Qi

Fowler School of Engineering

Agenda

- **Final Project**
- Functions: Default arguments
- In-Class Exercise

Final Project: Adventure Game

- Final Project takes ~50% of "Assignments", which takes 25% of the overall
- Details refer to "**Final_Project_Guideline**" on Canvas
- **Submission due: November 30th (Sunday) 23:59 PM**
  - **NO Extension would be administered.**
- **Final Presentation: December 1st (Monday), in-class**

Final Project: Best Practice

- **Start earlier, start earlier, start earlier!**
- **Divide and Conquer**
  - Divide the big project into modules
  - Handle module by module (puzzle)
  - ensure each module (function) works as expected:
    - If you have an input, ask yourself what the output should be.
    - Compare that with your code's outcome.
  - Once each module works as expected, connect them together and work on the logic of the overall.

Final Presentation

- **Randomly draw names** to decide the presentation order
- Must use **the slides submitted to Canvas**
- **5 minutes max** per person
- **Skip excessive details**
  - Summarize technical methods briefly
  - Avoid walking through code
  - Highlight key takeaways or unique solutions
- **Highlight challenges**
  - **Share 1 – 2 major challenges** you faced and how you tackled them
- **Discuss improvements**
  - Reflect on changes you'd make
  - Propose new features to make the game more fun

Functions II

Default arguments

Recap: Python Functions

- You **must** define a function before using it.

```
# Define a function to calculate the area of a rectangle
    # input parameters: length and width
    # output: area
def calculate_area(length, width):
    area = length * width # output of the function
    return area

# Use the function to calculat the area of rectangle1: 5(L) x 3(W)
length1 = 5
width1 = 3
area1 = calculate_area(length=length1, width=width1)
print("Area of rectangle 1 is", area1)
```

Define a function

Invoke (use) a function

Recap: Python Functions

- **Define a function**: Inputs are called ***parameters***
- **Invoke a function**: inputs are called ***arguments*** - specific values plugged into function parameters

parameters

```
def calculate_area(length, width):
    area = length * width # output of the function
    return area

# Use the function to calculat the area of rectangle1: 5(L) x 3(W)
length1 = 5
width1 = 3
area1 = calculate_area(length=length1, width=width1)
print("Area of rectangle 1 is", area1)
```

arguments

Default Arguments

- Default arguments allow functions to have **default values for parameters** within **the function definition**.
- When invoking the function, default arguments will be used **if no arguments are plugged in**.

Default Arguments

- Example:

"Guest" is the **default argument** for the parameter *name*

```
def greet(name = "Guest"):
    print("Welcome,", name, "!")

greet()      # Output: Welcome, Guest!
greet("Nicole")      # Output: Welcome, Nicole!
```

When invoking a function,

- If **no argument** is plugged in, the default argument will be used.
- If **an argument** is plugged in, that argument will be used.

Multiple Default Arguments

- A function can have **none, one, or multiple default arguments**.

```
def calculate_area(length=1, width=1):
    area = length * width
    return area

print(calculate_area())      # Output: 1
print(calculate_area(length=5))      # Output: 5
print(calculate_area(length=5, width=3))        # Output: 15
```

  - Both ‘width’ and ‘length’ have a default value of 1
  - You can provide both, one, or neither of them when calling the function.
  - **Good Practice**: **parameter = argument** when invoking a function for clarity, e.g., *calculate_area*(**length=5**, **width=3**)

Multiple Default Arguments

- However, if **NOT ALL parameters have default arguments**, ensure the ones with **default arguments are put at last**.

```python
def calculate_area(length, width=1):
    area = length * width
    return area
```

```python
def calculate_area(length = 1, width):
    area = length * width
    return area
```

In-Class Exercise

- Download the code from GitHub: https://github.com/trudiQ/cpsc230-Fall25-Qi
- Open **Lectures** -> Download **Functions_ForLoops_13_BLANK.py**
- Save the file to a folder named “cpsc230” on your desktop