CoGrammar

Week 10 – Open Class 1

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Software Engineering Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
  (FBV: Mutual Respect.)
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: **Open Class Questions**

Software Engineering Lecture Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**
- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**
- We would love your **feedback** on lectures: **Feedback on Lectures**

Progression Criteria

✓ Criterion 1: Initial Requirements

- Complete 15 hours of Guided Learning Hours and the first four tasks within two weeks.

✓ Criterion 2: Mid-Course Progress

- Software Engineering: Finish 14 tasks by week 8.
- Data Science: Finish 13 tasks by week 8.

✓ Criterion 3: Post-Course Progress

- Complete all mandatory tasks by 24th March 2024.
- Record an Invitation to Interview within 4 weeks of course completion, or by 30th March 2024.
- Achieve 112 GLH by 24th March 2024.

✓ Criterion 4: Employability

- Record a Final Job Outcome within 12 weeks of graduation, or by 23rd September 2024.

Lecture Objectives

1. Expand on functions by exploring Higher Order Functions.
2. Apply HOF to improve data management in programs.
3. Open Floor Q&A

Poll:

Assessment

What is a key characteristic of Higher-Order Functions (HOFs)?

A. They can only take and return functions as arguments.

B. They operate exclusively on numeric data types.

C. They can only be defined within classes.

D. They can accept functions as arguments and/or return functions.

What is a common use case for Higher-Order Functions?

A. Defining class attributes and methods.

B. Implementing conditional statements in functions.

C. Working with functions as first-class citizens.

D. Handling exceptions in try-except blocks.

First Class Functions

When a function is considered first-class, it indicates that the language views it as a value, allowing you to pass it around and assign it to variables. It is seldom used to describe a function, as in "a first-class function." Saying "a language has/hasn't got first-class function support" is far more common. Thus, a language has a “first-class functions” attribute.

What is a Higher order Function (HOF)

- ★ A function that takes other functions as arguments (or returns a function) is called a higher order function.
- ★ Higher-order functions operate by accepting a function as an argument, altering it, and then returning the altered function. More modular and reusable code can be produced as a result.

Power of HOF:

- **Adaptability when Transforming Data*:**
  Strong tools for handling and altering data are provided by functions like reduce(), filter(), and map(). They let you write operations clearly, often in just one line of code.

Power of HOF (continue):

- ★ reduce() – can lead to more concise and expressive code when dealing with operations that involve accumulating values over an iterable.
- ★ filter() – It filters elements from an iterable (e.g., a list) based on a specified condition.
- ★ map() – It applies a given function to all items in an iterable (e.g., a list) and return a new iterable containing the results.

Power of HOF (continue):

- **Lambda Operations (lambda <parameters>: <expression>):**
  Working with HOF is made even more concise by Python's support for lambda functions, also known as anonymous functions. You can write quick, disposable functions right within your code.

- **Using Iterables in Integration:**
  HOF integrates with Python's iterable objects with ease. This makes them particularly useful for tasks involving collections like lists, tuples, and dictionaries..

Wrapping Up

First Class Functions

Recognizing the power of first-class functions in Python allows us to treat functions as versatile entities leading to elegant, functional and dynamic programming

Higher Order Functions

In conclusion, practicing with Higher Order Functions not only empowers you to create modular and reusable code, but demonstrates Python's support for functional programming paradigms, making your codebase more expressive and flexible.

CoGrammar

Thank you for joining