Welcome to this CoGrammar Task Walkthrough: Task 6

The session will start shortly...

Questions? Drop them in the chat.
We'll have dedicated moderators answering questions.

Software Engineering Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** throughout this session, should you wish to ask any follow-up questions.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Software Engineering Session 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**

Enhancing Accessibility: Activate Browser Captions

Why Enable Browser Captions?

- Captions provide **real-time text for spoken content**, ensuring inclusivity.
- Ideal for individuals in noisy or quiet environments or for those with **hearing impairments**.

How to Activate Captions:

1. **YouTube or Video Players:**
   - Look for the CC (Closed Captions) icon and click to enable.
2. **Browser Settings:**
   - Google Chrome: Go to *Settings* > *Accessibility* > *Live Captions* and toggle ON.
   - Edge: Enable captions in *Settings* > *Accessibility*.

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Skills Bootcamp Progression Overview

✅ Criterion 1 - Initial Requirements

Specific achievements within the first two weeks of the program.

To meet this criterion, students need to, by no later than **01 December 2024:**

- **Guided Learning Hours** (GLH)**:** Attend a minimum of 7-8 GLH per week (lectures, workshops, or mentor calls) for a total minimum of **15 GLH**.
- **Task Completion:** Successfully complete the **first 4 of the assigned tasks**.

✅ Criterion 2 - Mid-Course Progress

Progress through the successful completion of tasks within the first half of the program.

To meet this criterion, students should, by no later than **12 January 2025:**

- **Guided Learning Hours** (GLH)**:** Complete at least **60 GLH**.
- **Task Completion :** Successfully complete the **first 13 of the assigned tasks**.

CoGrammar

Skills Bootcamp Progression Overview

✅ Criterion 3 – End-Course Progress

Showcasing students' progress nearing the completion of the course.

To meet this criterion, students should:

- **Guided Learning Hours** (GLH)**:** Complete the **total minimum required GLH,** by the **support end date.**
- **Task Completion : Complete all mandatory tasks**, including any necessary resubmissions, by the end of the bootcamp, **09 March 2025.**

✅ Criterion 4 - Employability

Demonstrating progress to find employment.

To meet this criterion, students should:

- **Record an Interview Invite:** Students are required to record proof of invitation to an interview by **30 March 2025**.
  - **South Holland Students** are required to proof and interview by **17 March 2025.**
- **Record a Final Job Outcome :** Within 12 weeks post-graduation, students are required to record a job outcome.

CoGrammar

Task Walkthrough: Programming with User-defined Functions

Learning Outcomes

- Define a Python function with parameters and return values.
- Execute two functions: one that takes input and returns a result, and another that performs a specific task based on input values.
- Explain the reasoning behind each code block and apply the logic to similar tasks.

Functions

- Functions allow you to encapsulate a set of instructions to be executed together, improving code readability and reusability.
- Function definition is done using the def keyword, followed by the function name and parentheses ().
- Arguments can be passed into functions to make them more flexible and reusable for different inputs.
- Return values are used to send results back to the part of the program that called the function.

Functions

- Built-in functions are provided by Python (e.g., print(), len()), while user-defined functions are created by you to perform specific tasks.

What are functions?

- Functions are reusable blocks of code that perform specific tasks.
- Called methods when used in OOP Classes.
- Functions help organise code, make it more readable, and facilitate debugging and maintenance.
- Useful for abstraction.
- Similarity to functions in maths, f(x) takes input x and produces some output.

Function Syntax

```
def function_name(parameters):
    """docstring"""
    # function body
    return statement
```

- **Function Name**: A unique identifier for the function.
- **Parameters** : Inputs to the function, specified within parentheses.
- **Function Body**: The block of code that performs the desired task.
- **Return Statement**: Optional statement that specifies the value returned by the function.

Return Statement

- In Python, the `**return**` statement is like sending a message back to where you called a function from.
- It's a way for the function to finish its job and share its final answer with the rest of the program.

Imagine you ask a friend to solve a math problem for you. After working on it, your friend comes back to you with the solution. In Python, the `return` statement is like your friend giving you that solution. It's the way the function tells the rest of the program what answer it found.

Part 1 Walkthrough

CoGrammar

Auto-graded task 1

1. Create a Python file called **holiday.py.**
2. Your task will be to calculate a user's total holiday cost, which includes the plane cost, hotel cost, and car rental cost.
3. First, get the following user inputs:
    - `city_flight`: The city they will be flying to (you can create some options for them. Remember, each city will have different flight costs).
    - `num_nights`: The number of nights they will be staying at a hotel.
    - `rental_days`: The number of days for which they will be hiring a car.
4. Next, create the following four functions:
    - `hotel_cost()`: This function will take `num_nights` as an argument and return a total cost for the hotel stay (you can choose the price per night charged at the hotel).

- `plane_cost()`: This function will take `city_flight` as an argument and return a cost for the flight. Hint: use **if/else** statements in the function to retrieve a price based on the chosen city.

- `car_rental()`: This function will take `rental_days` as an argument and return the total cost of the car rental (you can choose the daily rental cost).

- `holiday_cost()`: This function takes three arguments: `num_nights`, `city_flight`, and `rental_days`. Using these three arguments, call the `hotel_cost()`, `plane_cost()`, and `car_rental()` functions with their respective arguments, and finally return the total cost for the holiday.

5. Print out all the details about the holiday in a way that is easy to read.

Try running your program with different combinations of input to show its compatibility with different options.

holiday.py

Task Objective

The objective of this task is to demonstrate your ability to use Python functions, conditional logic, and user. You will:

- Write Python functions to calculate specific costs (e.g. flight, hotel and car rental costs).
- Use conditional logic within functions to vary costs based on user input.
- Dynamically combine and calculate values to determine the total cost.
- Output the holiday details in a clear and readable format.

This task will enhance your problem-solving skills and deepen your understanding of function definitions, argument passing, decision-making, and user interaction in Python.

Questions and Answers

CoGrammar

Documentation and Style

- Add comments to your code. Explain your approach, and/or how your code works.
- Consult the Python PEP8 guidelines: https://peps.python.org/pep-0008/

Pay close attention to:

- Variable names
- Spacing around operators
- Separating logical sections
- Indentation

```
# Define a variable to store the name of a user
user_name = "Alice"

# Print a greeting message using the user's name
print("Hello, " + user_name + "!")  # This prints: Hello, Alice!

# Define two numbers for basic arithmetic operations
num1 = 10
num2 = 5

# Calculate the sum of num1 and num2 and store the result in a variable
sum_result = num1 + num2
```

Learner Challenge

For those who are looking for an additional challenge, expand the functionality of your holiday cost calculator with the following features:

- Discount System:
  - Implement a discount system where:
  - If the total holiday cost exceeds a certain amount (e.g., £2000), the user gets a 10% discount.
  - For a longer stay (e.g., more than 7 nights), offer an additional hotel discount (e.g., £20 off per night beyond 7 nights).
- Multiple Destinations:
  - Allow the user to choose multiple cities for their trip. Update the plane_cost() function to handle multiple flights and calculate the total flight cost

Learner Challenge

- Flexible Car Rental:
  - Add a rule where users can rent a car only for specific days of their trip (e.g., they might not need the car for the entire stay). Update the program to calculate the rental cost based on their input.
- Error Handling:
  - Introduce error handling for user input, ensuring valid data is entered (e.g., numbers for num_nights and rental_days, valid city options, etc.). Provide helpful error messages for invalid inputs.
- Generate a detailed receipt showing:
  - Itemised costs for flights, hotel stays, car rentals, and add-ons.
  - Any applied discounts.

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar