Welcome to this CoGrammar Task Walkthrough: Task 8

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: <u>Questions</u>

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: <u>Feedback on Lectures</u>

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:

- 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



Rafig Manan

Scan to report a safeguarding concern



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



Ronald Munodawafa





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.





Learning Outcomes

- Explain the process of reading and writing files in Python.
- Apply for loops, list manipulation, and file operations in Python to solve problems that are similar to the walkthrough examples.
- Transfer your learnings to complete the tasks by the end of the session.



10 Operations

- File Operations:
 - Reading files: Opening files with open() in read mode ('r').
 - Writing files: Using open() in write mode ('w').
 - Context Managers: Explain the use of the with statement to manage file streams.
- Common String Operations:
 - o Splitting strings to separate data.
 - Stripping whitespace to clean input.
 - Looping through files line by line.



10 Operations

 Practical Use Cases: Storing data in lists and writing formatted output.



Part 1 Walkthrough





Auto-graded task 1

• Create a new Python file in the folder for this task, and call it **dob_task.py**.

 In your Python file, write a program that reads the data from the text file provided (DOB.txt) and prints it out in two different sections: one for names and another for birthdates, as shown in the format displayed

below:

Name

Orville Wright Rogelio Holloway Marjorie Figueroa ... etc.

Birthdate

21 July 1988 13 September 1988 9 October 1988 ... etc. Orville Wright 21 July 1988 Rogelio Holloway 13 September 1988 Marjorie Figueroa 9 October 1988 Debra Garner 7 February 1988 Tiffany Peters 25 July 1988 Hugh Foster 2 June 1988 Darren Christensen 21 January 1988 Shelia Harrison 28 July 1988 Ignacio James 12 September 1988 Jerry Keller 30 February 1988 Frankie Cobb 1 July 1988 Clayton Thomas 10 December 1988 Laura Reyes 9 November 1988 Danny Jensen 19 September 1988 Sabrina Garcia 20 October 1988 Winifred Wood 27 July 1988 Juan Kennedy 4 March 1988 Nina Beck 7 May 1988 Tanya Marshall 22 May 1988 Kelly Gardner 16 August 1988 Cristina Ortega 13 January 1988 Guy Carr 21 June 1988 Geneva Martinez 5 September 1988 Ricardo Howell 23 December 1988 Bernadette Rios 19 July 1988



dob_task.py - Task Objective

The objective of this task is to demonstrate your ability to work with file handling, data extraction, and formatted output in Python. You will:

- Create a new Python file named dob_task.py in the folder for this task.
- Read and extract data from the provided text file (DOB.txt).
- Organise the extracted data into two sections: 'Name' and 'Birthdate'.
- Format the output as per the format example provided.



Part 2 Walkthrough





Auto-graded task 2

Follow these steps:

- Create a file called student_register.py.
- Write a program that allows a user to register students for an exam venue.
- First, ask the user how many students are registering.
- Create a for loop that runs for that number of students.
- Each time the loop runs the program should ask the user to enter the next student ID number.
- Write each of the ID numbers to a text file called reg_form.txt.
- Include a dotted line after each student ID because this document will be used as an attendance register, which the students will sign when they arrive at the exam venue.



Student_register.py – Task Objective

The objective of this task is to demonstrate your understanding of user input, loops, and file handling in Python. You will:

- Create a new Python file named student_register.py.
- Ask the user how many students will be registering.
- Use a for loop to collect student ID numbers for the specified number of students.
- Save each ID number to a text file named reg_form.txt, ensuring:
 - Each student ID is followed by a dotted line, to act as a space for student signatures.



Questions and Answers





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
```



Lesson Conclusion

- File Operations:
 - o Reading files, Writing files, Context Managers
- Common String Operations:
 - o Splitting, Stripping, Looping



Learner Challenge - Option 1

Challenge 1: Advanced File Formatting

Objective: Build on the skills learned in reading and writing files.

- Modify the DOB.txt program (dob_task.py) to allow the user to filter and display data.
- Include an option for the user to:
 - 1. Display all names of people born in a specific month (e.g., "Show all people born in October").
 - 2. Sort the names alphabetically and print them in a separate section in the output.

Hint: Use string operations to extract the month from the birthdate and loops to filter the data.



Learner Challenge - Option 2

Challenge 2: Enhanced Student Registration

Objective: Extend the functionality of the student_register.py program.

- After registering the students and writing their IDs to reg_form.txt, allow the user to view or edit the data:
 - 1. Provide an option to display all registered student IDs.
 - 2. Allow the user to delete a specific student ID from the file by providing the ID.
 - 3. Save the changes back to reg_form.txt after editing.

Hint: Use a list to read and temporarily store the file contents for easier manipulation. Use file overwriting to save updates.



Questions and Answers





Thank you for attending





