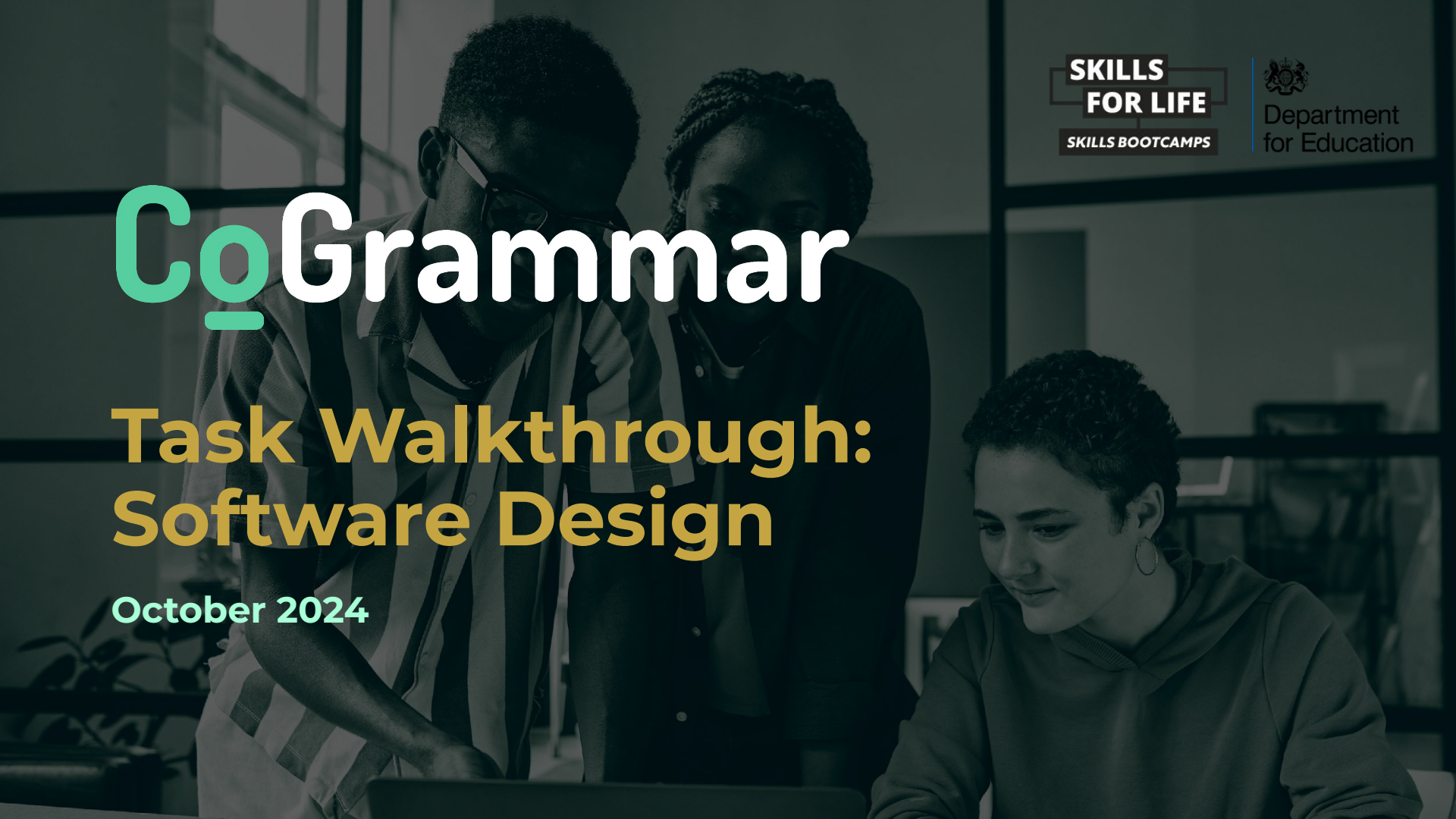
Welcome to this CoGrammar Tutorial:

Task Walkthrough

The session will start shortly...

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



SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

Task Walkthrough: Software Design

October 2024

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** 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 Academic Sessions. You can submit these questions here: **Questions**

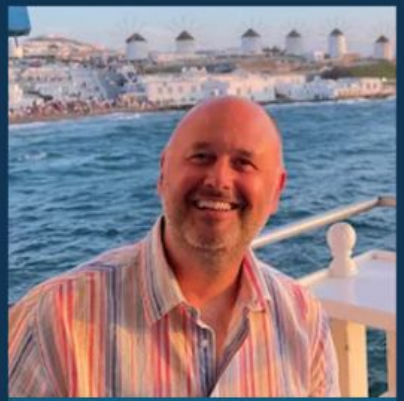
Software Engineering Session Housekeeping cont.

- 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**

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

Rafiq Manan

Charlotte Witcher

Nurhaan Snyman

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

To be eligible for a certificate of completion, students must fulfil three specific criteria. These criteria ensure a high standard of achievement and alignment with the requirements for the successful completion of a Skills Bootcamp.

✅ Criterion 1 - Meeting Initial Requirements

Criterion 1 involves specific achievements within the first two weeks of the program. To meet this criterion, students need to:

- Attend a minimum of 7-8 hours per week of guided learning (lectures, workshops, or mentor calls) within the initial two-week period, for a total minimum of 15 guided learning hours (GLH), by no later than 15 September 2024.
- Successfully complete the Initial Assessment by the end of the first 14 days, by no later than 15 September 2024.

Skills Bootcamp Progression Overview

✅ Criterion 2 - Demonstrating Mid-Course Progress

Criterion 2 involves demonstrating meaningful progress through the successful completion of tasks within the first half of the bootcamp.
To meet this criterion, students should:

- Complete 42 guided learning hours and the first half of the assigned tasks by the end of week 7, no later than 20 October 2024.

Skills Bootcamp Progression Overview

✅ Criterion 3 - Demonstrating Post-Course Progress

Criterion 3 involves showcasing students' progress after completing the course.
To meet this criterion, students should:

- Complete all mandatory tasks before the bootcamp's end date. This includes any necessary resubmissions, no later than 22 December 2024.
- Achieve at least 84 guided learning hours by the end of the bootcamp, 22 December 2024.

CoGrammar

Advised Resources

- HyperionDev PDF notes
- Lectures: 14, 16 & 17 October 2024
- Example code files
- Task walkthrough lecture
- Research

Learning Outcomes

- Identify and explain the purpose of a use case diagram, sequence diagram, and class diagram.
- Design and construct software design diagrams to show interactions between components in a system, user interactions, and the static structure of the system's classes.
- Apply the CRUD functionality and MVC pattern to create a system design.
- Demonstrate the ability to transfer learnings from software design principles by completing the Software Design task with 80% accuracy based on assessment criteria.

Software Design

- **Use Case Diagram:** Shows the system's functionality from the perspective of the user. It focuses on what the system does by listing the use cases (features) and actors (users).
- **Sequence Diagram:** Shows the sequence of messages exchanged between objects to perform a specific operation. Focuses on how the system behaves over time.
- **Class Diagram:** Shows the static structure of the system, focusing on the classes, their attributes, methods, and relationships.
- **MVC Components:**
  - Model: Represents the data and logic
  - View: Displays the user interface
  - Controller: Handles user input and updates the model and view
- **CRUD Matrices:**
  - Purpose: Identifying the operations (Create, Read, Update, Delete) needed for each entity

Task Walkthrough: Practical Task

CoGrammar

Practical task

In this task, you will review the design principles you have learned, and practise the skills you have been introduced to by **designing a task manager application** with features of your choice.

Need a quick and easy drawing tool? Check out **draw.io** for creating flowcharts, diagrams, and more. It's user-friendly and perfect for spicing up your projects visually. Give it a go and let your creativity shine!

Follow these steps:

- Create a use case diagram for your **task manager application**. You have creative freedom here, so your diagram can be as simple or as complex as you choose, based on the use cases you decide for your application. However, plan for your application to have the full range of CRUD (Create, Read, Update, Delete) functionality which should be evident within your diagrams.
- Create a sequence diagram for your task manager application. You may assume that your task manager application will utilise files to store its data.
- In a plain text file, clearly outline the specific responsibilities and concerns of each component (models, views, and controllers) in your task manager application, following the **MVC** (model-view-controller) pattern.
- Create a class diagram for your task manager application.
- Create a CRUD matrix for your task manager application.

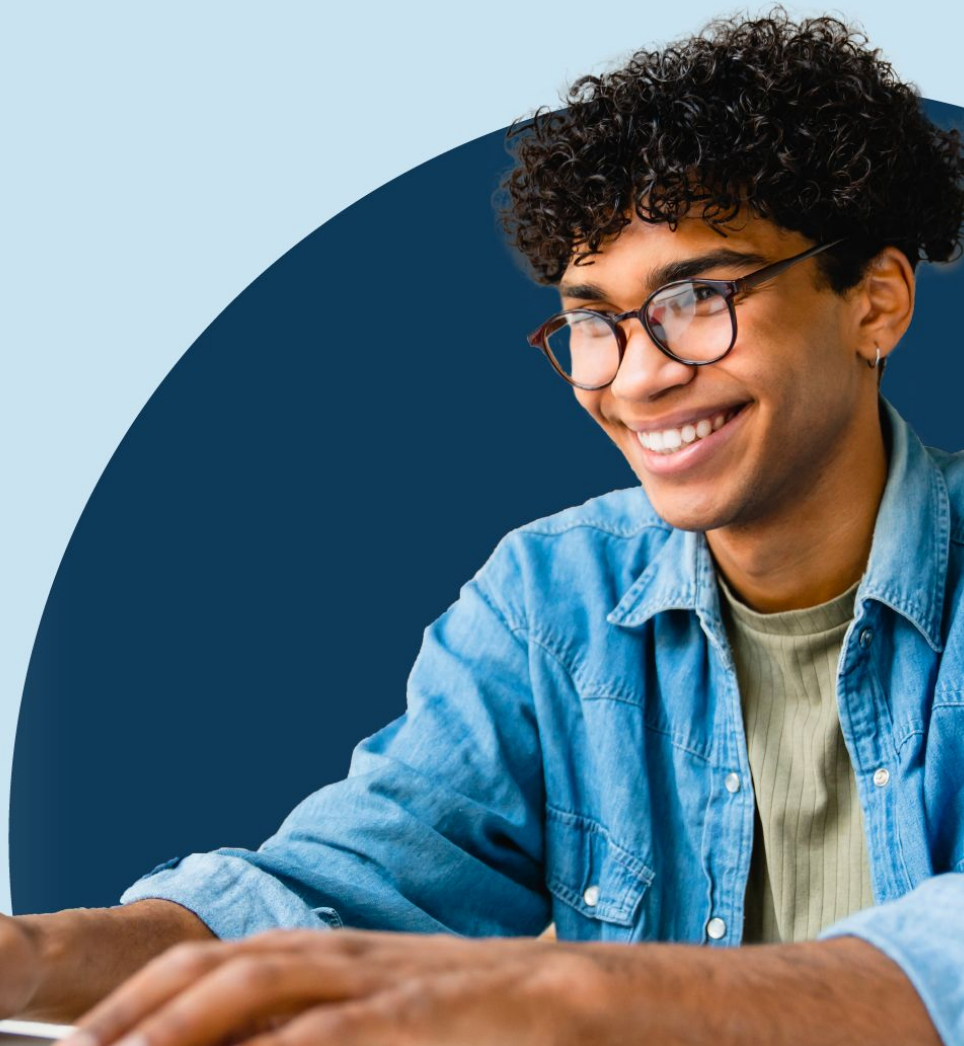
Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education



CoGrammar