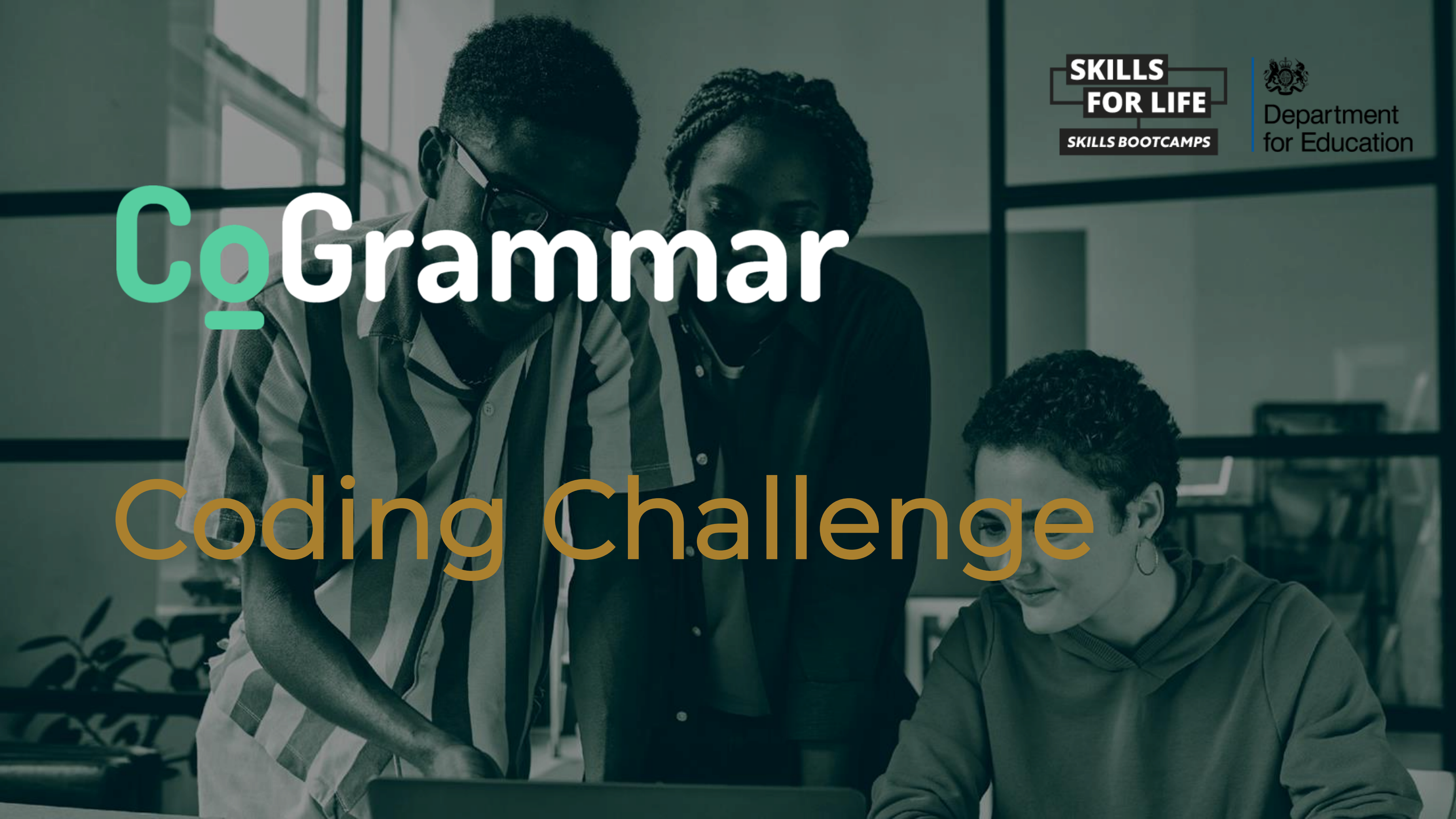
Welcome to the CoGrammar Face-to-face Coding Challenge

The session will start shortly...

Questions? Drop them in the chat.

CoGrammar



SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

Coding Challenge

Learning Outcomes

- Explain Core Banking Concepts
- Apply Programming Concepts to build a functional banking system
- Develop classes (UserAccount, BankSystem) using OOP principles such as encapsulation and data abstraction
- Analyse and Debug the Banking System
- Evaluate code quality and apply where code can be improved
- Demonstrate teamwork and present the project effectively

Coding Challenge

CoGrammar

Coding Challenge

Successfully build a **Basic Banking System**, by following a structured approach that ensures collaboration and leverages key programming concepts.

Coding Challenge

- Start by defining the core classes (e.g., User, Account, Transaction) using Object-Oriented Programming principles.
- Assign team members to work on specific components: e.g., one member or group of members can handle
    - *account creation and balance management*, another can
    - implement *deposit/withdrawal logic with conditionals*, and
    - a third can focus on *file handling to save transaction history*.

Coding Challenge

- Use loops for repetitive tasks like displaying account details or processing transactions.
- Ensure functions are created for modularity (e.g., deposit(), withdraw(), view_balance()).
- Regularly integrate individual contributions into a shared codebase, and test each feature as it's developed.

CoGrammar

Coding Challenge

- Finally, prepare a brief presentation highlighting the
  - implementation,
  - challenges faced, and
  - solutions applied.
- This approach ensures everyone contributes while reinforcing key skills.

CoGrammar

Questions and Answers

CoGrammar

Step Guide

- **Plan the system** (features, team roles)
- **Define core classes** (OOP design)
- **Implement banking operations** (menu system, transactions)
- **Handle file storage** (saving transaction history)
- **Test and debug** (fix errors, refine logic)
- **Integrate and optimise** (clean up code, add documentation)
- **Present the final project** (explain challenges & solutions)

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar