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.

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

Task Walkthrough: Object-Oriented Programming Inheritance | 2D Lists

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**

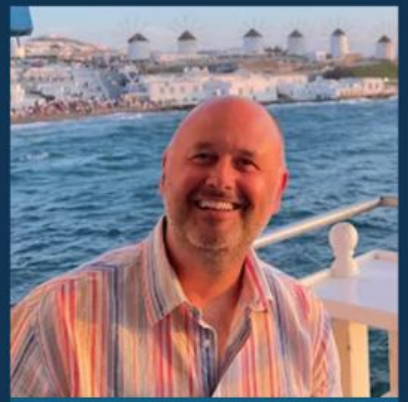
Software Engineering Session Housekeeping cont.

- 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.

CoGrammar

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.

CoGrammar

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: 30 September 2024 & 2, 3 October 2024 | Repeat on 4 & 5 October 2024
- Example code files
- Task walkthrough lecture
- Research

CoGrammar

Learning Outcomes

- Understand and implement Object-Oriented Programming concepts: Classes, objects, attributes, methods, inheritance and method overriding.
- Implement conditional logic for object instantiation.
- Explain how minesweeper-style algorithms work.
- Transfer your learnings to complete the tasks by the end of the session.

CoGrammar

OOP

- A **class** is a blueprint for creating objects, defining attributes and methods that objects from this class can use.
- An **object** is an instance of a class that contains actual values for the attributes defined by the class.
- **Attributes** are variables that belong to an object defined in the class's constructor using the __init__ method.
- **Methods** are functions defined inside a class that operate on instances of that class.
- A **constructor** is a special method that is automatically called when an object is created.
- **Inheritance** is a mechanism in object-oriented programming where a subclass inherits attributes and methods from a parent class.
- **Method overriding** occurs when a subclass provides a specific implementation of a method that is already defined in its parent class, allowing the subclass to customise or extend the behavior of that method.

Task Walkthrough: Auto-graded Task 1

CoGrammar

Auto-graded task 1

In this task, you will demonstrate your understanding of inheritance. Make a copy of the **task1_instructions.py** file and name it **practical_task_1.py.** Then, follow the instructions below.

- Add another method in the Course class that prints the head office location: Cape Town.
- Create a subclass of the Course `class` named `OOPCourse.`
- Create a constructor that initialises the following attributes with default values:
  - `description = “OOP Fundamentals”`
  - `trainer = "Mr Anon A. Mouse"`
- Create a method in the `OOPCourse` subclass named `trainer_details` that prints what the course is about and the name of the trainer by using the description and trainer attributes.
- Create a method in the `OOPCourse` subclass named `show_course_id` that prints the ID number of the course: `#12345`
- Create an object of the OOPCourse subclass called `course_1` and call the following methods
  - `contact_details()`
  - `trainer_details()`
  - `show_course_id()`
- These methods should all print out the correct information to the terminal.

CoGrammar

Task Walkthrough: Auto-graded Task 2

CoGrammar

Auto-graded task 2

Create a file named **method_override.py** and follow the instructions below:

- Take user inputs that ask for the name, age, hair colour, and eye colour of a person.
- Create an Adult class with the following attributes and method:
  - Attributes: `name, age, eye_color,` and `hair_color`
  - A method called `can_drive()` which prints the name of the person and that they are old enough to drive.
- Create a subclass of the `Adult` class named `Child` that has the same attributes, but overrides the `can_drive()` method to print the person's name and that they are too young to drive.
- Create some logic that determines if the person is 18 or older and create an instance of the `Adult` class if this is true. Otherwise, create an instance of the `Child` class. Once the object has been created, call the `can_drive()` method to print out whether the person is old enough to drive or not.

Be sure to place files for submission inside your **task folder** and click "**Request review**" on your dashboard.

CoGrammar

Task Walkthrough: Auto-graded Task 3

CoGrammar

Auto-graded task

Now it's time to see whether you're ready to apply what you've learned to some coding of your own! This is a challenging task, but worth persisting through as you'll gain valuable experience with 2D lists and nested loops.

1. Create a file named **minesweeper.py**.
2. Create a function that takes a grid of # and -, where each hash (#) represents a mine and each dash (-) represents a mine-free spot.
3. Return a grid where each dash is replaced by a digit, indicating the number of mines immediately adjacent to the spot, i.e., horizontally, vertically, and diagonally.

Example of an input:

```
[ ["-", "-", "-", "#", "#"],
  ["-", "#", "-", "-", "-"],
  ["-", "-", "#", "-", "-"],
  ["-", "#", "#", "-", "-"],
  ["-", "-", "-", "-", "-"] ]
```

Example of the expected output:

```
[ [1, 1, 2, "#", "#"],
  [1, "#", 3, 3, 2],
  [2, 4, "#", 2, 0],
  [1, "#", "#", 2, 0],
  [1, 2, 2, 1, 0] ]
```

CoGrammar

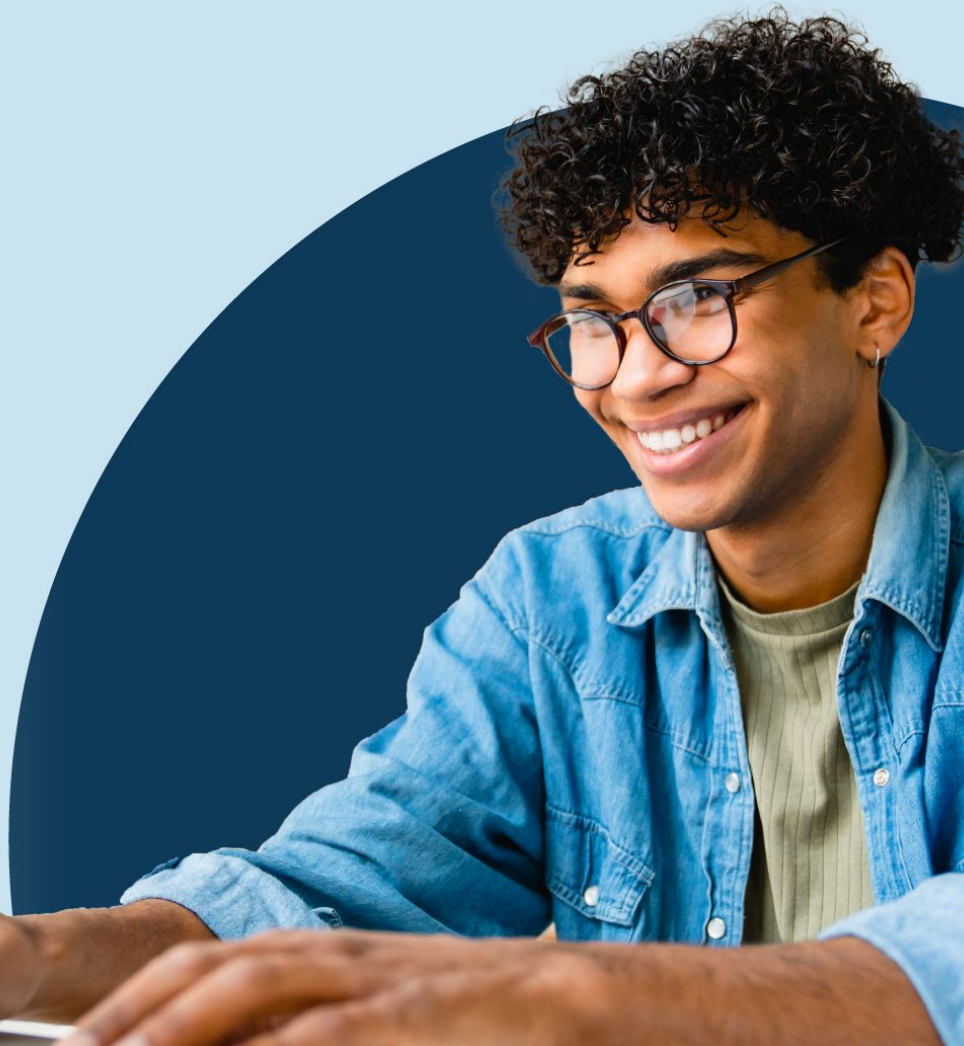
Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education



CoGrammar