Welcome to the CoGrammar

Coding Interview Workshop Portfolio Assignments

The session will start shortly...

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

Coding Interview Workshop 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]()**

Coding Interview Workshop 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**

Skills Bootcamp
8-Week Progression Overview

Fulfil 4 Criteria to Graduation

✅ Criterion 1: Initial Requirements

Timeframe: First 2 Weeks
Guided Learning Hours (GLH): Minimum of 15 hours
Task Completion: First four tasks

Due Date: 24 March 2024

✅ Criterion 2: Mid-Course Progress

60 Guided Learning Hours

Data Science - **13 tasks**
Software Engineering - **13 tasks**
Web Development - **13 tasks**

Due Date: 28 April 2024

CoGrammar

Skills Bootcamp
Progression Overview

✅ Criterion 3: Course Progress

Completion: All mandatory tasks, including Build Your Brand and resubmissions by study period end
Interview Invitation: Within 4 weeks post-course
Guided Learning Hours: Minimum of 112 hours by support end date (10.5 hours average, each week)

✅ Criterion 4: Demonstrating Employability

Final Job or Apprenticeship Outcome: Document within 12 weeks post-graduation
Relevance: Progression to employment or related opportunity

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

CIW Portfolio Assignments

2024

Portfolio Assignment Reviews

Submit you solutions here!

CoGrammar

Week 1

Sets, Variables, Functions and Order Complexity

CoGrammar

Portfolio Assignment: SE

Prime Number Finder Application

Objective: Create a simple application in Python that identifies prime numbers within a specified range. The application should allow users to input two numbers, representing the start and end of a range, and then output all the prime numbers within that range.

Portfolio Assignment: SE

Requirements:

- ➢ Use Python to develop the application.
- ➢ Incorporate functions to identify prime numbers and to handle user input.
- ➢ Demonstrate the use of integer data types and basic operations (unions, intersection).
- ➢ Provide comments to explain the logic behind prime number determination and any usage of set operations.
- ➢ Include a README file that explains how to run the application, the purpose of the program, and a brief description of the algorithm's time complexity.

Portfolio Assignment: DS

Data Filtering and Organisation Tool

Objective: Develop a Python script that filters and organises a dataset based on specific criteria. For instance, given a CSV file containing information about various individuals (e.g., name, age, occupation), the script should allow filtering for specific age ranges and occupations, then organising the output by name or age.

CoGrammar

Portfolio Assignment: DS

Requirements:

- Utilise Python, demonstrating the use of lists, sets, and dictionaries to manage and process data.
- Implement functions to filter data based on criteria and to sort the filtered data.
- Showcase the ability to read from and write to files, enabling the script to process input data and output results.
- Explain the use of modular arithmetic or greatest common divisors if applicable to data organisation.
- Accompany the script with a README file detailing how to use the tool, the types of data it can process, and a simple example with a sample dataset.

Portfolio Assignment: WD

Interactive Number Theory Playground

Objective: Create a web-based interactive tool that allows users to explore various number theory concepts (prime numbers, greatest common divisors, modular arithmetic) and set operations (union, intersection, difference, complement). This tool should enable users to input numbers and select the operation or concept they wish to explore, displaying the results on the webpage.

Portfolio Assignment: WD

Requirements:

- ➢ Use HTML/CSS to design a user-friendly interface for inputting values and selecting operations.
- ➢ Implement JavaScript to handle user inputs, perform the operations, and display the results dynamically on the page.
- ➢ Demonstrate type conversion and variable assignment.
- ➢ Include explanations or tooltips that briefly describe each number theory concept or set operation for educational purposes.
- ➢ Provide a README file with instructions on how to setup and use the web tool, along with an overview of the implemented features.

Week 2

Linear Algebra

Portfolio Assignment

- Feel free to adjust according to your interests and specific track (data science, web development, or software engineering)

CoGrammar

Portfolio Assignment

- Choose one or more of the following linear algebra topics to focus on:
  - Matrix operations (addition, subtraction, multiplication)
  - Linear transformations (rotation, scaling, shearing)
  - Solving systems of linear equations
  - Eigenvalues and eigenvectors

Portfolio Assignment

- Design and implement a web app that allows users to interact with the chosen topics. Some ideas include:
  - A matrix calculator that performs basic operations on user-inputted matrices
  - A 2D or 3D graphics visualiser that applies linear transformations to shapes based on user input
  - A linear equation solver that demonstrates the steps involved in Gaussian elimination

Portfolio Assignment

- An eigenvalue/eigenvector calculator that visualises the effect of eigenvalues on vector transformations

Portfolio Assignment

- Use HTML, CSS, and JavaScript to build the frontend of the app, focusing on creating a clean and intuitive user interface
- Implement the necessary mathematical calculations and algorithms using JavaScript or a backend language of your choice (e.g., Python, Java)
- Provide clear instructions on how to use the app and explain the linear algebra concepts being demonstrated

Portfolio Assignment

- Host the project on GitHub Pages or a similar platform and include a brief description of the app in the README file

Portfolio Assignment

- Evaluation Criteria:
  - Correctness and accuracy of the linear algebra calculations and visualisations
  - User interface design and ease of use
  - Code quality, organisation, and documentation
  - Clarity and effectiveness of the project description and instructions
  - Creativity and originality in applying linear algebra concepts to practical problems

Week 3

Counting, Probability and Statistics

CoGrammar

Portfolio Assignment: SE

Event Probability Simulator

Objective: Build a Python application that simulates the probability of different outcomes for given scenarios. The tool should allow users to define a scenario, including the number of trials and the specific events to simulate, and then output the probability of each event occurring.

CoGrammar

Portfolio Assignment: SE

Requirements:

- ➢ Develop the application in Python, using libraries like numpy or scipy for mathematical operations.
- ➢ Allow users to input parameters for simulations, such as the number of trials and event probabilities.
- ➢ Implement functionality to calculate permutations and combinations where needed to simulate scenarios (e.g. drawing cards from a deck, dice rolls).
- ➢ UtiliSe Bayes' theorem to update probabilities based on new information for certain scenarios.
- ➢ Include a README file that explains how to use the application, with examples of different scenarios and how to interpret output.

CoGrammar

Portfolio Assignment: DS

Descriptive Statistics and Data VisualiSation Tool

Objective: Create a Python script that computes descriptive statistics for a given dataset and visualizes these statistics through various charts and graphs. The tool should help in understanding the distribution, central tendency, and variability of data.

Portfolio Assignment: DS

Requirements:

- ➢ Utilize pandas for data manipulation and matplotlib or seaborn for creating visualizations.
- ➢ Compute basic descriptive statistics (mean, median, mode, standard deviation, quartiles) and display them in a user-friendly format.
- ➢ Create visualizations such as histograms, box plots, and scatter plots to represent the data distribution and statistical summaries.
- ➢ Implement functionality to apply basic inferential statistics techniques, like calculating confidence intervals for sample means.
- ➢ Provide a README file detailing the tool's functionality, how to run it, and examples of output given sample datasets.

Portfolio Assignment: WD

Interactive Hypothesis Testing Tool

Objective: Develop a web application that allows users to perform hypothesis testing on data they input. The application should guide users through selecting a hypothesis, choosing a significance level, and then inputting or uploading data to test against the hypothesis.

Portfolio Assignment: WD

Requirements:

- ➢ Build the application using HTML/CSS for the frontend and JavaScript for the backend logic.
- ➢ Design an intuitive UI for users to input their hypothesis, select a significance level (e.g., 0.05, 0.01), and input or upload their data.
- ➢ Calculate and display the result of the hypothesis test, including the p-value and whether the hypothesis is accepted or rejected at the chosen significance level.
- ➢ Offer brief explanations or tooltips about the hypothesis testing process and the significance of the p-value and confidence intervals.
- ➢ Include a README file with instructions on setting up and using the application, along with a simple example to demonstration.

Week 4

Number Bases, Binary Logic and Bitwise Operations (Compressions)

CoGrammar

Portfolio Assignment

- Feel free to adjust according to your interests and specific track (data science, web development, or software engineering)

Submit your solutions here!

CoGrammar

Portfolio Assignment: Data Science

Choose one or more of the following topics to focus on:

- Find elements in an array using XOR.
- Find repeating elements in an array of size n
- Find all odd occurring elements in an array

- Check if binary representation of a number is palindrome or not.

- Swapping variables without using a third variable (shown in this lecture)

Portfolio Assignment: Web development

- Color channel values are simply integers that range from 0 (0x00) to 255 (0xFF). Using bitwise operators (a) check whether a palette has the blue pixel or not, (b) insert a new value (0xEE) into the green channel of an existing color palette.

Portfolio Assignment: Software development

- A small car has 3 doors: 2 front doors (X, Y) and a hatchback (Z). The internal light of the car (L) is on when any of the car doors is open. Create and test a logic gates circuit to control the internal light (L) of the car based on whether a door (X, Y, or Z) is open (1) or closed (0).

Portfolio Assignment: Software development

- A bitmask is nothing more than a number that defines which bits are on and off, or a binary string representation of the number.
- Bitmask dynamic programming uses bitmasks, to keep track of our current state in a problem.
- Travelling salesman problem: Given a list of N cities and the distances between each pair of cities, what is the shortest possible route that the salesperson visits each city exactly once and returns to the city that they started from?
- Can be done using permutations, which slows down for large N.
- Use bitmasks and a recursive function instead.

CoGrammar

Portfolio Assignment

- Use HTML, CSS, and JavaScript to build the frontend of any app, focusing on creating a clean and intuitive user interface
- Implement the necessary mathematical calculations and algorithms using JavaScript or a backend language of your choice (e.g., Python, Java)
- Provide clear instructions on how to use the app and explain the concepts being demonstrated
- Host the project on GitHub Pages or a similar platform and include a brief description of the app in the README file

Submit your solutions here!

Portfolio Assignment

- Evaluation Criteria:
  - Correctness and accuracy of the calculations and visualisations
  - User interface design and ease of use
  - Code quality, organisation, and documentation
  - Clarity and effectiveness of the project description and instructions
  - Creativity and originality in applying concepts to practical problems

Week 5

Databases and SQL

CoGrammar

Portfolio Assignment: SE

Library Management System

Objective: Develop a library management system using SQL and Python which allows library staff to interact with the system to log loans, returns and new books being added to the system.

CoGrammar

Portfolio Assignment: SE

Requirements:

- ➢ Develop the application in Python, using libraries like mysql for SQL connections.
- ➢ Design a database schema for a library management system, including tables for books, members, and loans.
- ➢ Implement SQL queries to manage books, members, and loans.
- ➢ Write SQL queries to search for books, track loans, and manage library inventory.
- ➢ Implement a command-line or GUI interface for library staff to interact with the system.
- ➢ Include a README file that explains how to use the application, with examples of different scenarios and how to interpret output.

Portfolio Assignment: DS

Analyzing Customer Purchase Data

Objective: Apply SQL skills to analyse customer purchase data from an e-commerce database, focusing on CRUD operations, data manipulation, and database interactions within Python applications. Use SQL queries to calculate relevant metrics based on the data like the total revenue and average order value.

CoGrammar

Portfolio Assignment: DS

Requirements:

- ➢ Use Python's mysql module to connect to the database and execute SQL queries.
- ➢ Set up a database with tables for customers, products, and orders.
- ➢ Populate the database with sample data.
- ➢ Implement SQL queries to create, read, update, and delete records in the database.
- ➢ Write Python code to generate reports based on the analysis. This can include graphs and other visualisations as well.
- ➢ Write SQL queries to calculate total revenue, average order value, and other relevant metrics.
- ➢ Provide a README file detailing the tool's functionality, how to run it, and examples of output given sample datasets.

Portfolio Assignment: WD

Building an Online Marketplace

Objective: Create an online marketplace that interacts with a database using SQL and JavaScript. Include tables for products, sellers, and orders. Develop an attractive frontend using HTML for buyers to browse through products and place orders.

Week 6

Stacks and Queues + Function Objects and the Memory Model (Stack and Heap) (Pass by Reference and Value)

CoGrammar

Portfolio Assignment: SE

Project: Scheduler to manage task execution

Objective: Consider a simple scheduler designed to manage the execution of tasks of varying priority. We will consider two different implementations, one using a stack and another which uses a priority queue.

Provide a README file, how to run it, and examples of output.

Portfolio Assignment: SE

Requirements:

1. Discuss the choice between a stack or a priority queue for this implementation based on each data structure's performance and flexibility.

2. Based on the answer in 1, determine under what conditions it would be better to use each structure.

3. Implement a simple scheduler using either a stack or a priority queue.

CoGrammar

Provide a README file, how to run it, and examples of output.

Portfolio Assignment: DS

Project: Cafeteria queues and stacks

Objective: The school cafeteria serves brown (0) and white (1) bread toasted sandwiches to students. Students queue to get their preferred bread from the sandwiches kept in a stack. If the student at the front of the queue prefers the type of sandwich that is on top of the sandwich stack, they take the sandwich and leave the queue. Otherwise, they pass on the sandwich and move to the end of the queue. This continues until none of the queue students want to take the top sandwich and are thus unable to eat. Return the number of students that are unable to eat.

Provide a README file, how to run it, and examples of output.

Portfolio Assignment: WD

Project: Matching HTML tags

Objective: Uses stacks to match HTML tag in an HTML document. A simple opening HTML tag has the form “<name>” and the corresponding closing tag has the form “</name>”. Write a program to use stack to match the HTML tag and validate the input file.

CoGrammar

Provide a README file, how to run it, and examples of output.

Week 7

Recursion (including Stack Overflows) + Object-Oriented Programming (including Runtime Polymorphism)

Portfolio Assignment: SE

Object-Oriented Programming Design Patterns Implementation

Objective: Implement a set of design patterns using object-oriented programming principles in Python. This project will demonstrate your understanding of OOP and design patterns.

CoGrammar

Portfolio Assignment: SE

Requirements:

- ➢ Choose at least three design patterns from the following: Singleton, Factory Method, Observer, Strategy, or Composite.
- ➢ Implement the selected design patterns in Python, focusing on clean and modular code.
- ➢ Integrate the design patterns into a sample application (e.g., a simple game, a data processing tool).
- ➢ Provide documentation for each implemented design pattern, explaining its purpose and usage.
- ➢ Test the sample application to verify that the design patterns are functioning as expected.

Portfolio Assignment: DS

Recursive Neural Network for Text Classification

Objective: Implement a recursive neural network (RNN) in Python for text classification tasks. This project will showcase your understanding of recursion, OOP, and natural language processing (NLP) techniques.

Portfolio Assignment: DS

Requirements:

➢ Implement a recursive neural network using Python, utilising libraries such as TensorFlow or PyTorch.

➢ Use a publicly available text classification dataset (e.g., sentiment analysis, topic classification).

➢ Use OOP principles to organise the code and represent the neural network architecture.

➢ Train the RNN model on the dataset and evaluate its performance using appropriate metrics.

➢ Provide detailed documentation of the implementation, including the dataset used, model architecture, training process, and evaluation results.

Portfolio Assignment: WD

Interactive Recursive Tree Visualizer

Objective: Create a web application that allows users to visualise and interact with recursive tree structures. This project will demonstrate your understanding of recursion and frontend development skills.

Portfolio Assignment: WD

Requirements:

- Implement a recursive function in JavaScript to generate tree structures.
- Use HTML, CSS, and JavaScript to create an interactive visualization of the generated trees.
- Allow users to modify the tree structure (e.g., add/remove branches) through user interactions.
- Utilize OOP principles in JavaScript to represent tree nodes as objects.
- Update the visualization dynamically as users modify the tree structure.
- Provide clear documentation on how to use the application.

Week 8

Linear Data Structures (Singley + Doubley LL) + String and Character Encoding

CoGrammar

Portfolio Assignment: SE

Linked List Library

Objective: Develop a library for efficient manipulation of linked lists in Python or JavaScript, providing an easy-to-use interface for creating, manipulating, and querying linked lists, optimised for performance and thoroughly tested.

CoGrammar

Portfolio Assignment: SE

Requirements:

- ❖ Implement a linked list library in Python or JavaScript that provides an easy-to-use interface for creating, manipulating, and querying linked lists.
- ❖ Include support for both singly and doubly linked lists, with methods for insertion, deletion, searching, and traversal.
- ❖ Optimise the library for performance, considering the time complexity of each operation.
- ❖ Write unit tests to ensure the correctness of the library's functionality.
- ❖ Create a README file with documentation on how to install, use, and contribute to the library.

Portfolio Assignment: DS

Character Encoding Detector

Objective: Develop a Python script that detects the character encoding of text samples in a given dataset, analyses the distribution of different encodings, and discusses the implications of character encoding on data cleaning, preprocessing, and analysis in data science projects.

Portfolio Assignment: DS

Requirements:

- Create a Python script that takes a dataset containing text data as input and detects the character encoding of each text sample.
- Use libraries like chardet or cChardet to perform encoding detection.
- Analyse the distribution of different character encodings in the dataset and visualise the results using matplotlib or seaborn.
- Discuss the implications of character encoding on data cleaning, preprocessing, and analysis in the context of data science projects.
- Provide a Jupyter Notebook with well-documented code, explanations, and insights.

Portfolio Assignment: WD

Web-Based Text Converter

Objective: Create a web application that allows users to input text and convert it between different character encodings, providing an intuitive user interface and error handling for unsupported characters or encoding-related issues.

CoGrammar

Portfolio Assignment: WD

Requirements:

- Create a web application using HTML, CSS, and JavaScript that allows users to input text and convert it between different character encodings (e.g., ASCII, UTF-8, UTF-16).
- Implement the encoding conversion logic using JavaScript or a server-side language like Python or Node.js.
- Design an intuitive user interface that displays the input text, allows selecting the source and target encodings, and shows the converted output.
- Include error handling for unsupported characters or encoding-related issues.
- Deploy the web application on a platform like GitHub Pages or Heroku and provide a link to the live demo in the README file.

Week 9

Hash Tables

CoGrammar

Portfolio Assignment: SE

Implementing a Hash Table for Symbol Table Management

Objective: Develop a command-line tool in Python for managing symbol tables using a hash table data structure. The tool should allow users to insert, delete, and retrieve symbols, and handle collisions using appropriate techniques.

Portfolio Assignment: SE

Requirements:

➢ Implement functions for inserting symbols into the hash table, deleting symbols, and retrieving symbols based on their names.

➢ Provide a command-line interface for interacting with the symbol table, including options for inserting, deleting, and retrieving symbols.

➢ Include error handling for cases where symbols cannot be inserted or retrieved due to collisions or other issues.

➢ Provide a README file that explains how to use the tool, including examples of valid commands and their expected outputs.

Portfolio Assignment: DS

Analyzing Movie Ratings using a Hash Table

Objective: Develop a Python script that analyzes movie ratings data using a hash table. The script should read a dataset containing movie ratings, store the ratings in a hash table, and perform various analyses such as finding the average rating per movie and identifying the highest rated movies.

Portfolio Assignment: DS

Requirements:

- Read a CSV file containing movie ratings data and populate a hash table with this data.
- Implement functions to calculate the average rating per movie and identify the highest rated movies based on the ratings stored in the hash table.
- Provide a command-line interface for running the script and displaying the analysis results.
- Include error handling for cases where the input file is not found or the data is not valid.
- Provide a README file that explains how to run the script and includes a sample input file.

Portfolio Assignment: WD

Implementing a Hash Table for URL Shortening

Objective: Develop a web application that shortens long URLs using a hash table data structure. The application should allow users to input a long URL, generate a shortened URL using the hash table, and redirect to the original URL when the shortened URL is accessed.

Portfolio Assignment: WD

Requirements:

- Implement the web application using HTML, CSS, and JS.
- Implement a hash table data structure in JavaScript to store the mappings between shortened URLs and original URLs.
- Use a hash function to generate hash codes for the original URLs and handle collisions appropriately.
- Include error handling for cases where the shortened URL does not exist or when the original URL is not valid.
- Provide a README file that explains how to run the web application locally. You may use React.js to manage packages, dependencies and creating a server.

Week 10

Trees (including Heaps)

CoGrammar

Week 11

Graphs

CoGrammar

Portfolio Assignment: SE

Network Routing Simulator

Objective: Develop a command-line tool in Python that simulates network routing using graph algorithms.

Portfolio Assignment: SE

Requirements:

- Implement a graph class in Python to represent the network.
- Allow users to input network data in a standard format.
- Simulate packet routing between nodes, demonstrating the shortest path and alternative routes.
- Include error handling for disconnected nodes and unreachable destinations.
- Provide clear output showing the steps of the routing process and results.
- Include a README file explaining how to use the tool.
- Provide example inputs and outputs for different network scenarios.

Portfolio Assignment: DS

Social Network Analysis

Objective: Analyze a real-world social network dataset to extract meaningful insights using graph theory.

CoGrammar

Portfolio Assignment: DS

Requirements:

- ➢ Obtain a social network dataset.
- ➢ Clean and preprocess the data to construct a graph where nodes represent users and edges represent connections.
- ➢ Analyze the structure of the graph (e.g., degree distribution, clustering coefficient).
- ➢ Perform centrality analysis to find key influencers.
- ➢ Identify communities and clusters within the network.
- ➢ Write a report detailing the methodology, analysis, and insights.
- ➢ Include visualizations and code snippets in the report.
- ➢ Provide a README file showing how to visualise the analysis.

Portfolio Assignment: WD

Interactive Graph Visualizer

Objective: Develop an interactive web application that visualizes graphs and demonstrates graph traversal algorithms.

CoGrammar

Portfolio Assignment: WD

Requirements:

- Use HTML, CSS, and JavaScript to create the user interface.
- Allow users to draw and edit graphs by adding/removing vertices and edges.
- Implement visual representations for different types of graphs (directed, undirected, weighted).
- Implement a JavaScript backend that processes graph data.
- Use an appropriate graph library for graph operations.
- Include error handling for invalid graph inputs.
- Provide interactive examples within the application.
- Include a README file explaining how to use the application.
- Provide example graphs and use cases.

Week 12

Parallelism, Concurrency and Scheduling

CoGrammar

Portfolio Assignment Reviews

Submit you solutions here!

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar