Introduction to Software Development Week 11: Revision & Next Steps

1. Learning Objectives

- Consolidate understanding of all core concepts covered in the module.
- Review the connections between topics (e.g., how functions, collections, and OOP work together).
- Identify key areas for further learning in software development.
- Prepare for the final project.

2. Core Concepts Revision

- **From Idea to Code:** The SDLC (Planning -> Design -> Implementation -> Testing).
- **The Building Blocks:**
  - **Data:** Variables, Types (`int`, `str`, `bool`), Collections (`list`).
  - **Logic:** Operators, Control Flow (`if`, `for`, `while`).
- **Structuring Code:**
  - **Functions:** For reusability and modularity (DRY principle).
  - **Classes/Objects (OOP):** To model real-world entities by bundling data (attributes) and behaviour (methods).
- **Ensuring Quality:**
  - **Error Handling (`try...except`):** Building robust code that doesn't crash.
  - **Debugging:** The process of finding and fixing logical errors.
  - **Unit Testing:** Verifying that individual components work correctly.
- **Managing Code:**
  - **Git & GitHub:** For tracking changes and collaborating.

3. The Big Picture: How It All Connects

Imagine building a simple "User" system.

- You use a **Class** `User` to define the blueprint (OOP).
- Each `User` object has **attributes** like `name` and `email` (**Data**).
- It might have a `login()` **method** that contains **control flow** (`if` password is correct...).
- You might store multiple `User` objects in a **list** (**Collection**).
- You would write **unit tests** to ensure the `login()` method works.
- You'd use **Git** to save every change you make to the `User` class.
- Your `login()` method should use **`try...except`** to handle potential network errors.

4. Next Steps in Your Journey

- **Languages & Frameworks:** Explore other languages (Java, C#, JavaScript) or specialise with Python frameworks (Django for web, Pygame for games).
- **Databases:** Learn SQL or NoSQL to persist data permanently.
- **Web Development:** Understand Front-End (HTML, CSS, JavaScript) and Back-End (server-side logic).

- **Data Structures & Algorithms:** Deeper study of efficient ways to store and process data.