Project 1 Question 2 documentation

The use of functions in the re-factored code provides several advantages:

1. Modularity: By encapsulating the pattern printing logic in a separate function, the code becomes more modular and easier to understand. The `printPattern` function has a specific purpose and can be reused or modified independently of the rest of the code.

2. Code Re-usability: The `printPattern` function can be called multiple times with different inputs, allowing you to print patterns for various numbers without duplicating code. This promotes code reuse and avoids unnecessary repetition.

3. Readability and Maintainability: The use of functions can improve the readability and maintainability of the code. The main function becomes cleaner and more focused on the high-level logic, while the detailed pattern printing logic is delegated to the `printPattern` function. This separation of concerns makes the code easier to understand, debug, and modify in the future.

4. Encapsulation: The `printPattern` function encapsulates the pattern printing logic, keeping it separate from the input/output operations in the `main` function. This promotes encapsulation, which is a fundamental principle of good software design. It helps with code organization, reduces potential naming conflicts, and makes it easier to reason about each component of the program independently.

Overall, using functions helps improve the structure, modularity, re-usability, and maintainability of the code. It allows you to break down complex tasks into smaller, more manageable parts and promotes good software engineering practices.