Chapter 5: Formatting

The Purpose of Formatting

Formatting is super important for programmers because it's how we make sure everyone can read and understand the code. Even though getting the code to work is the main goal, making it look nice and organized is just as important. Good formatting helps everyone on the team understand the code better.

Vertical Formatting

Vertical formatting is about how you organize the code from top to bottom. Some tips include:

Blank Lines: Use blank lines to separate different parts of your code, like functions or sections, so it's easier to read.

Order of Functions: Put related functions close to each other in a logical order, so it's clear how everything fits together.

Horizontal Formatting

Horizontal formatting is about how wide your code lines are. Some tips include:

Line Length: Keep lines of code shorter than 120 characters so you don't have to scroll sideways to read them.

Spaces: Use spaces around operators (like + or =) to make the code easier to read.

Alignment: Only line things up when it makes the code clearer. Too much alignment can be distracting.

Team Rules

When you're working with a team, you all need to follow the same formatting rules. Even if the team's style is different from yours, it's important to stick to the team rules so that the code looks consistent and everyone can read it easily.

Chapter 6: Objects and Data Structures

Data Abstraction

Data abstraction means hiding the details of how data is stored and only showing what's necessary. This helps keep the code simple and makes it easier to change how data is handled without breaking other parts of the code.

Data/Object Anti-Symmetry

There are key differences between data structures and objects:

Data Structures: These are simple and show their data directly without many functions.

Objects: These hide their data and provide methods to work with that data, which makes the code more organized and flexible.

The Law of Demeter

The Law of Demeter is a rule that says objects should only talk to their immediate friends. This means an object should avoid calling methods on objects returned by other methods. Following this rule makes the code less complicated and easier to change.

Data Transfer Objects (DTOs)

DTOs are simple objects used to move data between different parts of a system, like between a database and the code. They help keep the internal data separate from the external data, making it easier to handle data changes.