W02 Assignment: Explain Abstraction

Natalie Kolbo

Abstraction is a principle in programming that simplifies complex systems by focusing on essential details and hiding the implementation details. For example, when we use the print() function in Python, we don't need to know how it interacts with the operating system; we just call it to display text.

One benefit of abstraction is that it reduces complexity and makes code easier to maintain. By focusing on what a program does rather than how it works, abstraction allows developers to reuse and modify components independently.

In the journal program, the Entry class abstracts the details of a single journal entry, such as the date, prompt, and response. It also provides a Display method that encapsulates how the entry is presented to the user. Similarly, the Journal class abstracts the management of multiple entries. Its AddEntry method provides a clean way to add entries, while the Display method handles the presentation of all entries. This ensures that changes to the internal workings of these classes do not affect other parts of the program.

Here is a code snippet demonstrating abstraction:

csharp

public void AddEntry(Entry entry)

{

\_entries.Add(entry);

}

public void Display()

{

foreach (Entry entry in \_entries)

{

entry.Display();

}

}

This approach simplifies the program and ensures that the logic for storing and displaying entries is encapsulated within the respective classes, making the code modular and easier to maintain.