NAME: ADEBIMPE ADEROJU M.

COURSE: CSE210- PROGRAMMING WITH FUNCTIONS

INSTRUCTOR'S NAME: JOHN READING

What is abstraction and why is it important?

Abstraction means simplifying complex systems by focusing only on essential features while hiding unnecessary details for example the print("Hello World") function in python has more than 3000 lines of code but only shows this short form as an abstraction. In programming, this creates simple interfaces for complex operations—like pressing "power" on a TV remote without understanding the internal circuits.

Key Benefits:

- **Reduces cognitive load**—users learn simple method calls instead of complex internals
- **Easy maintenance**—change implementation without breaking calling code
- **Team collaboration**—developers work independently on different components

Real Application: Journal Class
In the Journal program, the Journal class perfectly demonstrates abstraction. Users call simple methods like AddEntry() and SaveToFile("data.txt") without knowing how entries are stored or files are handled.

```
public class Journal
{
   private List<Entry> _entries = new List<Entry>();

   public void AddEntry(Entry newEntry)
   {
      _entries.Add(newEntry);  // Hides list management
   }

   public void SaveToFile(string filename)
   {
      // Hides complex file I/O, serialization logic
      using (StreamWriter writer = new StreamWriter(filename))
```

```
        {
            foreach (Entry entry in _entries)
                writer.WriteLine(entry.ToCsvString());
        }
    }
}
```

Why This is Important: Program.cs just calls journal.SaveToFile("journal.txt"). Tomorrow, I could swap file storage for database storage—same method call, zero changes elsewhere. The Resume.Display() from W02 worked identically, hiding the foreach loop complexity. Abstraction = "trust the black box."