Core Functionality

1. **Library (DLL) Creation:**
    - Create a dynamic link library (DLL) containing all the class definitions.
    - Ensure the classes are instantiated and used polymorphically to output attributes.
2. **Inheritance Hierarchy:**
    - **Create abstract class `Analyzer`:** Base class for all analyzers.
    - **Create `PrintAnalyzer`:** Subclass that finds fingerprints ('@') in a grid.
    - **Create `SampleAnalyzer`:** Subclass that collects DNA/hair/fiber samples in a grid.
    - **Create `ThirdAnalyzer`:** Inherit from `SampleAnalyzer`, with distinct methods or data.
    - **Polymorphism:** Implement methods to ensure all analyzers can be used polymorphically.

User Input & File Handling

1. **File Selection:**
    - Allow the user to enter/select the crime scene data file.
    - Design the input format and structure of the file (e.g., text, CSV).
    - Implement file reading functionality (use provided sample code from D2L as a reference).
    - Ensure proper error handling for file input issues (invalid format, file not found).
2. **Data Storage in Object Variables:**
    - Parse the file data and store it in the appropriate class variables of the analyzers.

UI Design (Labels, Text Boxes, Buttons)

1. **Form UI:**
    - Design the form interface with labels, text boxes, and buttons.
2. **User Inputs:**
    - Text box for entering the filename (case number or name).
    - Buttons for selecting the type of evidence to examine (fingerprints, DNA, etc.).

Evidence Collection & Display

1. **Result of Evidence Collection:**
    - Display the results of the fingerprint collection as a string (number of '@' symbols found).
    - Display the results of other evidence collected as a dynamically created 2D array of picture boxes (using `.png` files for visualization).
    - Optional: Make boundary lines of the 2D array visible or invisible based on the design.
2. **Display Messages:**
    - Display a success message when evidence collection is complete.
    - Display the number of fingerprints or samples collected.
3. **Image Handling:**
    - Find or create appropriate `.png` files for the evidence and display them in the picture boxes.

Exception Handling

1. **Handle Input Errors:**
    - Use a message box to handle exceptions (e.g., file not found, file format errors).
    - Ensure proper validation and error handling at each step.

Additional Features

1. **Switch Statement:**
    - Implement a `switch` statement to handle the different types of analyzers based on user input.