# HexDisplay Program Design Documentation

## Design Methodology: Top-Down Approach

The HexDisplay program is designed using the **top-down design methodology**, where the problem is broken down into smaller, manageable components. Each component is commented briefly in the code, but the comments can demonstrate the flow of information between these components.

## Program Structure: Tree Diagram

Below is the tree diagram representing the structure of the program:

Main Function

├── checkHexFile

├── readAndInsertData

│   ├── Parse Hex Line

│   ├── Validate Checksum

│   └── Handle Record Types

│       ├── Data Record (Type 0)

│       ├── Extended Linear Address Record (Type 4)

│       └── End of File Record (Type 1)

└── displayHexData

   ├── Print Hexadecimal Values

   └── Print ASCII Representation

## Explanation of Information Flow

1. **Main Function**

-  **Responsibility**: Acts as the entry point of the program. It initializes the data array, opens the input file, and calls other functions to process and display the hex data.

-  **Information Flow**:

   -  Receives the input file name from the command line.

   -  Passes the file pointer and data array to other functions.

2. **checkHexFile**

-  **Responsibility**: Validates the input file to ensure it exists and has a .hex extension.

-  **Information Flow**:

   -  Receives the file pointer and file name.

   -  Outputs an error message and exits if the file is invalid.

3. **readAndInsertData**

-  **Responsibility**: Reads the hex file line by line, parses the data, validates the checksum, and stores the data in the totalByteData array.

-  **Information Flow**:

   -  Receives the file pointer and the totalByteData array.

   -  Parses each line of the hex file to extract:

      -  **Byte Count**: Number of data bytes in the record.

      -  **Address**: Starting memory address for the data.

      -  **Record Type**: Type of the record (e.g., data, extended address, end of file).

      -  **Data**: Actual data bytes.

      -  **Checksum**: Validates the integrity of the record.

   -  Updates the totalByteData array with the parsed data.

4. **displayHexData**

-  **Responsibility**: Displays the contents of the totalByteData array in both hexadecimal and ASCII formats.

-  **Information Flow**:

   -  Iterates through the totalByteData array.

   -  Prints 16 bytes per line in hexadecimal format.

   -  Converts each byte to its ASCII equivalent (if printable) and displays it alongside the hex values.