C++ Coding Question: Inheritance & Dynamic Memory Management

Problem Statement: Library Management System

You need to design a simple **Library Management System** using **C++ inheritance** and **dynamic memory allocation**. The system should have the following classes:

Class: `Book` (Base Class)

Represents a general book with basic details.

Attributes:

- `string title` – Title of the book.
- `string author` – Author of the book.
- `int publicationYear` – Year of publication.

Methods:

- `Book(string title, string author, int year)` – Constructor to initialize a book.
- `virtual void displayInfo() const` – Displays book details (to be overridden in derived classes).
- `virtual ~Book()` – Virtual destructor.

Class: `EBook` (Derived Class from `Book`)

Represents an electronic book that has additional attributes.

Attributes:

- `string* formats` – A dynamically allocated array that stores multiple formats (e.g., PDF, EPUB, MOBI).
- `int formatCount` – The number of formats available.

Methods:

- `EBook(string title, string author, int year, int formatCount, string* formatList)` – Constructor that initializes an eBook and dynamically allocates formats.
- `void displayInfo() const override` – Displays eBook details including formats.
- `~EBook()` – Destructor to deallocate memory.

Class: `Library` (Manager Class)

Manages a collection of books and eBooks.

Attributes:

- `Book** books` – Dynamically allocated array of `Book*` (stores both `Book` and `EBook` objects).
- `int bookCount` – Number of books in the library.

Methods:

- `Library()` – Default constructor initializing an empty library.
- `void addBook(Book* newBook)` – Adds a book (either `Book` or `EBook`) to the library.
- `void removeBook(int index)` – Removes a book at a given index and shifts elements.
- `void displayAllBooks() const` – Displays details of all books.
- `~Library()` – Destructor to clean up dynamically allocated memory.

Task

Implement all the above classes with their respective attributes and methods in C++. Ensure:

- **Proper use of inheritance**
- **Dynamic memory allocation and deallocation (Destructor implementation)**
- **Manager class to handle CRUD operations (Create, Read, Update, Delete)**

You may use the following main function to test your implementation:

```
int main() {
    Library library;

    // Creating books
    Book* b1 = new Book("The Alchemist", "Paulo Coelho", 1988);
    Book* b2 = new Book("1984", "George Orwell", 1949);

    // Creating eBooks with dynamic formats
    string formats1[] = {"PDF", "EPUB"};
    EBook* e1 = new EBook("Clean Code", "Robert C. Martin", 2008, 2,
formats1);

    // Adding books to library
    library.addBook(b1);
    library.addBook(b2);
    library.addBook(e1);

    // Display all books
    library.displayAllBooks();
```

```
    // Remove a book
    library.removeBook(1);

    // Display again
    library.displayAllBooks();

    return 0;
}
```

Expected Output (Example):

```
Book: The Alchemist, Author: Paulo Coelho, Year: 1988
Book: 1984, Author: George Orwell, Year: 1949
EBook: Clean Code, Author: Robert C. Martin, Year: 2008, Formats: PDF, EPUB

Removing book at index 1...

Book: The Alchemist, Author: Paulo Coelho, Year: 1988
EBook: Clean Code, Author: Robert C. Martin, Year: 2008, Formats: PDF, EPUB
```

Extra Challenge (Optional)

- Implement an `updateBook()` method in `Library` to modify book details.
- Allow users to search for a book by title.
- Implement **deep copy** constructor and **copy assignment operator** for `EBook` to properly handle dynamic memory.

This problem will test your knowledge of:

- **Inheritance**
- **Dynamic memory allocation (`new` / `delete`)**
- **Virtual destructors**
- **CRUD operations with manager classes**