C++ Coding Question: Grooming Store Management System

Problem Statement:

You need to design a simple **Grooming Store Management System** using C++ **inheritance** and **dynamic memory allocation**. The system should handle pets (like dogs and cats), their grooming appointments, and a manager class to handle CRUD operations.

Class: `Pet` (Base Class)

Represents a general pet with basic details.

Attributes:

- `std::string name` – Name of the pet.
- `std::string breed` – Breed of the pet.
- `int age` – Age of the pet.

Methods:

- `Pet(std::string name, std::string breed, int age);`
  - Constructor to initialize a pet.
- `virtual void displayInfo() const;`
  - Displays pet details (to be overridden in derived classes).
- `virtual ~Pet();`
  - Virtual destructor.

Class: `Dog` (Derived Class from `Pet`)

Represents a dog that requires additional attributes related to grooming.

Attributes:

- `bool needsHaircut;` – Indicates if the dog needs a haircut.
- `std::string* vaccinations;` – Dynamically allocated array storing vaccination names.
- `int vaccineCount;` – Number of vaccinations.

Methods:

- `Dog(std::string name, std::string breed, int age, bool needsHaircut, int vaccineCount, std::string* vaccineList);`

    - Constructor that initializes a dog and dynamically allocates vaccinations.
- `void displayInfo() const override;`
    - Displays dog details including haircut need and vaccinations.
- `~Dog();`
    - Destructor to deallocate memory.

Class: `Cat` (Derived Class from `Pet`)

Represents a cat that requires additional attributes related to grooming.

Attributes:

- `bool needsNailTrim;` – Indicates if the cat needs a nail trim.
- `std::string* favoriteToys;` – Dynamically allocated array storing favorite toys.
- `int toyCount;` – Number of toys.

Methods:

- `Cat(std::string name, std::string breed, int age, bool needsNailTrim, int toyCount, std::string* toyList);`
    - Constructor that initializes a cat and dynamically allocates favorite toys.
- `void displayInfo() const override;`
    - Displays cat details including nail trim need and favorite toys.
- `~Cat();`
    - Destructor to deallocate memory.

Class: `GroomingStore` (Manager Class)

Manages a collection of pets.

Attributes:

- `Pet** pets;` – Dynamically allocated array of `Pet*` (stores both `Dog` and `Cat` objects).
- `int petCount;` – Number of pets in the store.

Methods:

- `GroomingStore();`
    - Default constructor initializing an empty store.
- `void addPet(Pet* newPet);`
    - Adds a pet (`Dog` or `Cat`) to the store.
- `void removePet(int index);`

    - Removes a pet at a given index and shifts elements.
- `void displayAllPets() const;`
    - Displays details of all pets.
- `~GroomingStore();`
    - 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)**

Example `main()` Function

```
int main() {
    GroomingStore store;

    // Creating dogs with vaccinations
    std::string vaccines1[] = {"Rabies", "Parvo"};
    Dog* dog1 = new Dog("Buddy", "Golden Retriever", 3, true, 2, vaccines1);

    std::string vaccines2[] = {"Rabies"};
    Dog* dog2 = new Dog("Max", "Beagle", 2, false, 1, vaccines2);

    // Creating cats with favorite toys
    std::string toys1[] = {"Feather Wand", "Laser Pointer"};
    Cat* cat1 = new Cat("Whiskers", "Siamese", 4, true, 2, toys1);

    std::string toys2[] = {"Ball", "Scratching Post"};
    Cat* cat2 = new Cat("Mittens", "Persian", 5, false, 2, toys2);

    // Adding pets to the store
    store.addPet(dog1);
    store.addPet(dog2);
    store.addPet(cat1);
    store.addPet(cat2);

    // Display all pets
    store.displayAllPets();

    // Remove a pet
    store.removePet(1);

    // Display again
    store.displayAllPets();
    return 0;
}
```

Expected Output (Example)

```
Dog: Buddy, Breed: Golden Retriever, Age: 3, Needs Haircut: Yes,
Vaccinations: Rabies, Parvo
Dog: Max, Breed: Beagle, Age: 2, Needs Haircut: No, Vaccinations: Rabies
Cat: Whiskers, Breed: Siamese, Age: 4, Needs Nail Trim: Yes, Favorite Toys:
Feather Wand, Laser Pointer
Cat: Mittens, Breed: Persian, Age: 5, Needs Nail Trim: No, Favorite Toys:
Ball, Scratching Post
Removing pet at index 1...
Dog: Buddy, Breed: Golden Retriever, Age: 3, Needs Haircut: Yes,
Vaccinations: Rabies, Parvo
Cat: Whiskers, Breed: Siamese, Age: 4, Needs Nail Trim: Yes, Favorite Toys:
Feather Wand, Laser Pointer
Cat: Mittens, Breed: Persian, Age: 5, Needs Nail Trim: No, Favorite Toys:
Ball, Scratching Post
```

Extra Challenge (Optional)

- Implement an `updatePet()` method in `GroomingStore` to modify pet details.
- Allow users to search for a pet by name.
- Implement **deep copy constructor** and **copy assignment operator** for `Dog` and `Cat` to properly handle dynamic memory.

Concepts Covered

This problem will test your knowledge of:

- **Inheritance**
- **Dynamic memory allocation (`new` / `delete`)**
- **Virtual destructors**
- **CRUD operations with a manager class**
- **Deep copy constructor & assignment operator (optional challenge)**