

# C++ Problem Statement: "Battle Game Character Manager"

---

You're building a **Character Management System** for a **fantasy battle game**. The game features multiple characters (warriors, wizards, etc.), and players can recruit, view, and train them. You will manage a few game characters using C++.

Your program must use **OOP, file handling, inheritance, operator overloading, exception handling, and pointers**.

---

## Class Requirements

### 1. Abstract Class: **Character**

- Members:
  - `char name[50]`
  - `int level`
  - `int HP`
- Functions:
  - Pure virtual: `void displayStats() = 0;`
  - Virtual destructor

### 2. Derived Classes: **Warrior, Wizard**

- Additional Members:
  - `int strength` (for Warrior)
  - `int mana` (for Wizard)
- Implement `displayStats()`

### 3. Class: **Player**

- Members:
  - `char playerName[50]`
  - `Character* team[3]` → max 3 characters
  - `int teamSize`
- Functions:
  - `void recruitCharacter(Character* c)`
  - `void showTeam()`

### 4. Operator Overloading

- Overload `<<` to print character stats (use friend function)
-

## File Handling Requirements

- Load initial characters from `characters.txt` on program start
- Each line has format:

```
Warrior,Thorin,5,150,80
Wizard,Gandalf,10,100,120
```

- Save the player's team to `team.txt` on exit

---

## Functional Requirements

Menu:

- ```
1. View Available Characters
2. Recruit a Character (by name)
3. View My Team
4. Exit
```

- Use a simple array to store available characters: `Character* available[10];`
- Use pointers for polymorphism
- Use basic exception handling (e.g., invalid character name, team full)

---

## Example Gameplay Flow

```
Welcome, Commander!

1. View Available Characters
2. Recruit a Character
3. View My Team
4. Exit

> 1
Available Characters:
[Warrior] Thorin - Level 5, HP 150, Strength 80
[Wizard] Gandalf - Level 10, HP 100, Mana 120

> 2
Enter character name to recruit: Gandalf
Gandalf recruited successfully!

> 3
Your Team:
[Wizard] Gandalf - Level 10, HP 100, Mana 120
```

---

 **Time Limit:** 30 minutes

---

☒ **Submission Instructions**

- Submit a `.cpp` file
  - Use arrays and pointers only
  - Use comments where needed
  - Make sure it reads `characters.txt` and saves to `team.txt`
- 

Bonus:

- Implement virtual destructor in `Character`
  - Use custom exception class like `class GameException : public exception`
-