



Task(17.3)

This Task simulates a basic gaming scenario where multiple Player objects share access to a GameSession but each player exclusively controls their own Character. The program demonstrates the use of smart pointers (`std::unique_ptr` and `std::shared_ptr`) to manage memory safely and efficiently.

Classes and Concepts:

1.Character Class:

- Represents a player's in-game character.
- When a Character is created, its name is passed to the constructor, and a message is displayed indicating the creation.
- When a Character object is destroyed, a message is printed to indicate that.
- The class also has an attack method, which outputs that the character is attacking.

2.GameSession Class:

- Represents the overall game session that can be shared by multiple players.
- When the session starts, a message is printed, and when the session ends, another message is shown.
- It has a startSession method, which prints a message when the game session is active.

3.Player Class:

- Each Player object has a name, a `std::shared_ptr` to a GameSession (allowing shared access), and a `std::unique_ptr` to a Character (exclusive ownership). The Player class has a setCharacter method
- that assigns a Character to the player using a `std::unique_ptr`. The `std::move` operation is used to transfer ownership of the Character. The play method prints that the player is playing, calls the
- attack method on the character (if it exists), and starts the game session by calling startSession.

Key Features:

- Memory Management:
 - `std::shared_ptr<GameSession>`: The GameSession object is shared among all players. Multiple players can hold a reference to the session, and it will only be destroyed when all references to it go out of scope.
 - `std::unique_ptr<Character>`: Each player has exclusive ownership of their character. The Character is destroyed automatically when the Player object goes out of scope or when a new character is assigned.
- Behavior:
 - When the program runs, a GameSession is created and shared between two players (Player1 and Player2).
 - Each player creates their own character: Player1 has a "Warrior" and Player2 has a "Mage."
 - Both players "play" the game by attacking with their respective characters and starting the shared game session.

Thank You