

Do We Need Extra Data to Initialize Any Fields?

```
struct player_t {
    char name[32];
    char password[20];
    int32_t age;
    int32_t num_games;
    int32_t score_dist[16];
    struct game_t* game;
};
```

Pass into init function.

Initialize to 0 / NULL.

Writing the Player Initialization Function

```
int32_t player_init
(struct player_t* p,
 const char* name,
 const char* pswd, int32_t p_age)
{
    int32_t i;
    for (i = 0; 31 > i && '\0' != name[i];
        i++) {
        p->name[i] = name[i];
    }
    p->name[i] = '\0';
}
```

the player

name

password

age

Writing the Player Initialization Function

```
int32_t player_init
(struct player_t* p,
 const char* name,
 const char* pswd, int32_t p_age)
{
    int32_t i;
    for (i = 0; 31 > i && '\0' != name[i];
        i++) {
        p->name[i] = name[i];
    }
    p->name[i] = '\0';
}
```

Copy up to 31 characters.

Terminate with NUL.

Make Choices to Reduce Likelihood of Bugs

What happens

- in the function so far
- if one writes `name` instead of `p->name`?
- Or if one writes `p->name` instead of `name`?

The compiler can't help you.

Avoid using field names as arguments.

Then the compiler can help if you make a mistake.