

# UML Documentation

## Combatants

```
# Name: string # maxHitPoints: int  
# currentHitPoints: int + getSmithHitPoints() int +  
+ getMaxHitPoints() int + getName(): string  
+ getInitNumber(): int + TookDamage(damage: int): void
```

## Monster

```
- Weapon: weapons - initiative Mod: float  
- proficiency Bonus: int - Attack Bonus: int  
+ rollInitiative(): int + setInitiative(): void  
+ makeAttack(): void + tellDamage(): void  
+ display(): void + getWeapon(): weapons  
+ getProfBonus(): int + getAttack Bonus(): int  
+ getInitMod(): int + setInitMod(mod: int): void
```

## Player

```
+ setInitiative(): void  
+ enterCurrentHP(currentHitPoints: int): void  
+ display(): void
```

Combatents <del>Common like</del> Hit points: int, Defense: int,
--

Creation of objects Monsters, populate attributes Player hp tracking not this version
---

Knight, Thief  
 Druid

Display Initiative Order (Combatant InitArray[max])

InitArray[0]

↓ cout "These are the attributes."

InitArray[1]

cout "These are the attributes."

What if I add monsters & players to the array after determining initiative order?

The "Reverse array" assignment may be key to switching monsters in an array.

Does the Initiative array even need to be monsters only their names?

I'm going to have to have a player initiative array and a Monster Initiative array. Couldn't solve the problem in time.

Do more documentation pages.

I need to think of combining the two arrays for Monsters and Players into a final array of Combatant type.

Back to Initiative!

And we're back it Broke all the things  
Refer to the note above to fix. I need  
a break for now.



# Determine Order Initiative Brain storm

Initial

{Combatant1, Combatant2, Null/default, ..., Null/default}

↓  
loop through {Combatant1, ..., Combatant2, ..., Null/default, ..., Null/default}

Ver1. while (Array[~~##~~] != Null/default) {  
for(counter, counter < Max(Combat, counter++))

int index = 0;  
Ver2. while (Array[<sup>index</sup>##] != Null/default) {  
if (index < Max(Combatants)) {  
for(counter, counter < ~~index~~, counter++) {

switch order by overwriting the data of each  
index++

she said older and wiser  
not older and wiser.

# Program flow

