

Assignment Instructions

1. Characters and Classes

The game supports three character classes.

Each class has **default stats**, and stats are **not provided during character creation**.

Default Class Stats

Class	Health	Strength	Defense
-------	--------	----------	---------

Warrior	100	15	10
---------	-----	----	----

Thief	60	20	7
-------	----	----	---

Mage	40	25	5
------	----	----	---

Each character has an **inventory** for equipped items
(implementation is up to you).

Character Creation

```
create <class> <name>
```

2. Items and Potions

Weapon / Equipment Default Stats

Item	Health	Strength	Defense
Sword	0	5	10
Dagger	0	7	3
Wand	0	10	0
Shield	0	0	5

Potions

Potion	Effect
Health Potion	+10 Health
Strength Potion	+5 Strength
Defense Potion	+5 Defense

Potions are consumed automatically when picked up.
They do not go into the inventory.

3. Item Placement and Upgrades

Items can receive **additional stat bonuses** when placed.

```
place <item> <room> <hpBonus> <strBonus> <defBonus>
```

Example:

```
place sword Entrance 0 5 0
```

Creates a sword with:

- Strength: 10
- Defense: 10

To create a default item, use zeros:

```
place sword Entrance 0 0 0
```

4. Class Equipment Restrictions

- **Warrior** → Sword, Shield
- **Thief** → Dagger
- **Mage** → Wand

Attempting to equip an invalid item **fails**.

5. Rooms and Map Structure

Room Creation

```
create room <roomName>
```

Rooms are stored internally in a **linked list**.

Connecting Rooms

```
connect <roomA> <roomB> <direction>
```

Directions:

- North
- South
- East
- West

Rules:

- No two rooms may connect to the same room
- No circular connections are allowed
- Each room has a unique name

6. Dungeon Start

`Set StartRoom <roomName>`

`Enter Dungeon <characterName>`

- The player starts in the room defined by `Set StartRoom`
- Calling `Enter Dungeon` before setting a start room results in an error

7. Movement

`Move <characterName> <direction>`

- Movement is only possible if a connection exists
- Invalid movement prints an error and continues the game

8. Combat System

`Fight <characterName> <monsterName>`

Combat rules:

- Turn-based
- Player always attacks first
- Combat continues until one side dies

Damage Formula

`damage = max(1, attacker STR - defender DEF)`

Combat Result

- If the monster dies → **Victory**
- If the player dies → **Lose**

Health does NOT reset after combat.

9. Monsters

```
place monster <name> <room> <health> <strength> <defense>
```

Each monster exists in one room only.

10. Item Pickup Rules

```
PickUp <characterName> <itemName>
```

- If the item is not in the current room → print an error
- If the item is a potion → apply effect immediately
- If the item is equipment → attempt to equip it

Duplicate Equipment Rule

- If the character already has an item of the same type:
 - Compare **total stats**
 - Keep the item with **higher total stats**
 - If stats are equal → **keep the currently equipped item**

11. Output Requirements

- Output must describe **only actions and their results**
- Output format does **not** need to match exactly
- Meaning must be clear and consistent

Example Output

Thorin moves North to Hallway.

Thorin fights Goblin: Victory

Thorin tries to pick up Sword: Failed (item not in room).

12. Notes

- There is **no add game function**
- Only **create** and **connect** are used for setup
- Errors do **not** stop the game