CS 329E: Bulko
Programming Assignment 1
Role-Playing Games

1 Problem Definition

A *role-playing game* or RPG is a game in which players assume the roles of characters in a fictional setting. The popularity of the epic saga told in J.R.R. Tolkien's *The Hobbit* and *The Lord of The Rings* greatly influenced the genre, as seen in the game *Dungeons & Dragons* and all of its subsequent variants.

In most RPGs, a player creates a *character* of a specific archetype or *class*, such as “Fighter", “Wizard", or “Thief". (**Note:** do not confuse a “character class" with a Swift class; although we will define character classes using Swift classes, they do not mean the same thing, so it might be a little confusing.) The class defines a set of characteristics, constraints, and behavior that apply to all characters of that class. For example, a Thief has the ability to move with great stealth; however, in order to do that effectively, the Thief cannot wear bulky, noisy armor or carry large weapons.

In this assignment, you will create a very simple RPG. You will define two character classes, define characteristics and actions associated with those classes, and create one character of each class. You will define types of weapons and armor suitable for your characters to use. Finally, you will have your two characters battle each other!

2 Weapons:

Define a Swift class called **Weapon.** When a Weapon object is created, it should include as a parameter a String that identifies the weapon *type.* Each weapon type will do a specific amount of damage when it strikes an opponent.

Valid weapon types and the damage done by each are:

- “dagger": 4 points of damage
- “axe": 6 points of damage
- “staff": 6 points of damage
- “sword": 10 points of damage
- “none": bare hands; default 1 point of damage

3 Armor:

Define a Swift class called **Armor.** When an Armor object is created, it should include as a parameter a String that identifies the armor *type*. Each armor type offers a specific amount of protection (the “Armor Class" or AC) when an opponent attacks the character wearing the armor.

Valid armor types and the AC associated with each are:

- “plate": plate armor with AC 2
- “chain": chain mail with AC 5
- “leather": leather mail with AC 8
- “none": no armor worn; default AC 10

4 Characters:

Define a Swift class called **RPGCharacter**.

Define a subclass of RPGCharacter called **Fighter**. Fighters possess the following characteristics:

- The maximum Health a Fighter can have at any time is 40 points. If a character’s health drops below zero, the character is said to be “defeated".
- The maximum Spell Points a Fighter can have at any time is 0. Spell Points are used to cast magic spells; Fighters cannot use magic.
- Fighters are allowed to use any weapon type.
- Fighters are allowed to wear any type of armor.

Define a subclass of RPGCharacter called **Wizard**. Wizards possess the following characteristics:

- The maximum Health a Wizard can have at any time is 16 points.
- The maximum Spell Points a Wizard can have at any time is 20.
- Wizards can only fight with a “dagger", a “staff", or their bare hands.
- Wizards cannot wear armor. It interferes with their ability to cast spells.

When a character is created, it should include as a parameter a String that identifies the character’s name. Creation of a character should also initialize the following characteristics:

- No armor being worn.
- No weapon being wielded.
- Current Health equal to the maximum Health for its class.
- Current Spell Points equal to the maximum Spell Points for its class.

5 Actions:

A character of any class can perform the following actions:

- He/she can wield a weapon. Define a method `wield` that takes a weapon object as a parameter. If the weapon type is allowed for that character's class, the method should print the message, `"NAME is now wielding a(n) WEAPONTYPE"`; otherwise, it should print `"Weapon not allowed for this character class."`
- He/she can unwield a weapon. Define a method `unwield` that sets the character's current weapon to "none" and prints the message, `"NAME is no longer wielding anything."`
- He/she can put on armor. Define a method `putOnArmor` that takes an armor object as a parameter. If the armor type is allowed for that character's class, the method should print the message, `"NAME is now wearing ARMORTYPE"`; otherwise, it should print `"Armor not allowed for this character class."`
- He/she can take off armor. Define a method `takeOffArmor` that sets the character's current armor to "none" and prints the message, `"NAME is no longer wearing any armor."`
- He/she can fight another character. Define a method `fight` that takes a **character object** as a parameter. The method should:
  - print the message, `"NAME attacks OPPONENTNAME with a(n) WEAPONTYPE"`.
  - deduct the amount of damage done by the weapon type from the opponent's current health.
  - print the message, `"NAME does DAMAGE damage to OPPONENTNAME"`.
  - print the message, `"OPPONENTNAME is now down to CURRENTHEALTH health"`.
  - check to see if the opponent has been defeated. (See below.)

A wizard (and only a wizard) can also perform the following action:

- He/she can cast a spell. Define a method `castSpell` that takes as parameters a String (the Spell Name) and a **character object** (the target of the spell).
- A spell has a Spell Name, a *cost* in Spell Points, and an *effect* in Health Points. A spell is always targeted at a specific character, possibly even the caster him/herself. When the spell is cast, the caster consumes the cost in Spell Points, and the target is affected by the effect in Health.
- The three different spells available are:
  - `"Fireball"`: cost = 3, effect = 5.
  - `"Lightning Bolt"`: cost = 10, effect = 10.
  - `"Heal"`: cost = 6, effect = -6. (That means the target's Health *increases* by 6 points.)
- When a character casts a spell, the method should:
  - print the message, `"NAME casts SPELLNAME at TARGETNAME"`.
  - if the Spell Name is not one of the three listed, print the message, `"Unknown spell name. Spell failed."` and return.

  - if the number of Spell Points possessed by the caster is less than the cost of the spell, print the message, `"Insufficient spell points"` and return.
  - adjust the target's Health based on the effect of the spell. Note that casting a `Heal` spell can never raise the target's Health above its maximum.
  - adjust the caster's Spell Points based on the cost of the spell.
  - if the spell was a `Heal` spell, print the message, `"CASTER heals TARGET for ### health points"`, followed by `"TARGET is now at ### health."` Otherwise, print the message, `"CASTER does ### damage to TARGET"`, followed by `"TARGET is now down to ### health"`, and then check to see if the target was defeated. (See below.)

Two other methods you must define are:

- `show` for a character object should create a string and print it, so that `characterName.show()` would print out the following:

  Name of character
      Current Health: ##
      Current Spell Points: ##
      Wielding: "xxxxx"
      Wearing: "xxxxx"
      Armor Class: ##

- `checkForDefeat` should take a character object as a parameter, and if the character's current Health is less than or equal to zero, print the message, `"NAME has been defeated!"`

6 Main Program:

Test your program thoroughly with your own characters and sequence of actions. However, to make it easier to grade your assignment, turn in your code including the following main program:

```
// top level code

// Create one of each armor and weapon for use
let plateMail = Armor(armorType:"plate")
let chainMail = Armor(armorType:"chain")
let sword = Weapon(weaponType:"sword")
let staff = Weapon(weaponType:"staff")
let axe = Weapon(weaponType:"axe")

let gandalf = Wizard(name:"Gandalf the Grey")
gandalf.wield(weaponObject:staff)

let aragorn = Fighter(name:"Aragorn")
aragorn.putOnArmor(armorObject:plateMail)
aragorn.wield(weaponObject:axe)

gandalf.show()
```

```
aragorn.show()

gandalf.castSpell(spellName:"Fireball",target:aragorn)
aragorn.fight(opponent:gandalf)

gandalf.show()
aragorn.show()

gandalf.castSpell(spellName:"Lightning Bolt",target:aragorn)
aragorn.wield(weaponObject:sword)

gandalf.show()
aragorn.show()

gandalf.castSpell(spellName:"Heal",target:gandalf)
aragorn.fight(opponent:gandalf)

gandalf.fight(opponent:aragorn)
aragorn.fight(opponent:gandalf)

gandalf.show()
aragorn.show()
```

For debugging purposes, here is the expected output that this main program should produce:

```

Gandalf the Grey is now wielding a(n) staff
Aragorn is now wearing plate
Aragorn is now wielding a(n) axe
Gandalf the Grey
   Current Health:  16
   Current Spell Points:  20
   Wielding:  staff
   Wearing:  none
   Armor class:  10
Aragorn
   Current Health:  40
   Current Spell Points:  0
   Wielding:  axe
   Wearing:  plate
   Armor class:  2
Gandalf the Grey casts Fireball at Aragorn
Gandalf the Grey does 5 damage to Aragorn
Aragorn is now down to 35 health
Aragorn attacks Gandalf the Grey with a(n) axe
Aragorn does 6 damage to Gandalf the Grey
```

```
Gandalf the Grey is now down to 10 health
Gandalf the Grey
   Current Health:  10
   Current Spell Points:  17
   Wielding:  staff
   Wearing:  none
   Armor class:  10
Aragorn
   Current Health:  35
   Current Spell Points:  0
   Wielding:  axe
   Wearing:  plate
   Armor class:  2
Gandalf the Grey casts Lightning Bolt at Aragorn
Gandalf the Grey does 10 damage to Aragorn
Aragorn is now down to 25 health
Aragorn is now wielding a(n) sword
Gandalf the Grey
   Current Health:  10
   Current Spell Points:  7
   Wielding:  staff
   Wearing:  none
   Armor class:  10
Aragorn
   Current Health:  25
   Current Spell Points:  0
   Wielding:  sword
   Wearing:  plate
   Armor class:  2
Gandalf the Grey casts Heal at Gandalf the Grey
Gandalf the Grey heals Gandalf the Grey for 6 health points
Gandalf the Grey is now at 16 health
Aragorn attacks Gandalf the Grey with a(n) sword
Aragorn does 10 damage to Gandalf the Grey
Gandalf the Grey is now down to 6 health
Gandalf the Grey attacks Aragorn with a(n) staff
Gandalf the Grey does 6 damage to Aragorn
Aragorn is now down to 19 health
Aragorn attacks Gandalf the Grey with a(n) sword
Aragorn does 10 damage to Gandalf the Grey
Gandalf the Grey is now down to -4 health
Gandalf the Grey has been defeated!
Gandalf the Grey
   Current Health:  -4
   Current Spell Points:  1
   Wielding:  staff
   Wearing:  none
   Armor class:  10
```

```
Aragorn
   Current Health:  19
   Current Spell Points:  0
   Wielding:  sword
   Wearing:  plate
   Armor class:  2
```

Disclaimer for serious RPG gamers: this RPG has been stripped down for the purpose of keeping the homework assignment simple. Consequently, several standard features common to RPGs have been omitted; for example:

- “Armor class" is defined, but we don’t do anything with it.
- Battle is not random: attacks always hit and do the maximum damage.
- Spells always hit and do the maximum damage.
- Characters don’t have a formal inventory.
- Characters don’t have levels.

However, this doesn’t prevent you from adding these features yourself if you want to create your own fun RPG! Sorry, no extra credit for doing this, though. :-)

7 Grading criteria

1. The required classes are defined. (10%)
2. The required methods are defined in a way that accurately reflects object-oriented programming principles. (20%)
3. The program runs and produces the expected output. (70%)
4. Ensure the output of your program **closely** resembles the expected output. I am less concerned with exact formatting (spacing, field sizes, etc.) and more concerned with content (that your messages closely match what you were instructed to print).
5. The Coding Standard is followed. One point deducted for each violation. Exception: since you?ll be writing your code in the playground, you do not have to put all of your classes and structures in separate files.

8 General criteria

1. I will be looking for good documentation, descriptive variable names, clean logical structure, and adherence to all coding conventions expected of an experienced programmer, as well as those outlined in the Coding Standard document. There will be penalties for failure to meet these standards.
2. Xcode will automatically generate standard headers to your .swift files. Add two lines to each Swift file so that the header includes the following:

```
// Project: LastnameFirstname-HW1
// EID: xxxxxx
// Course: CS329E
```

9 To turn in your assignment:

As shown in the required heading above, name your file `<yourLastName><yourFirstName>-HW1`. Zip up the .playground folder; note that it looks like a file in the Finder, but it's actually a folder. Attach the final version of your zip file to the HW1 assignment on Canvas before the due date/time.