tbc module needs to…

import random

create a class called Character function

initialize using method \_\_init\_\_(self)

begin with super().\_\_init\_\_()

make a variable called self.name, give it the value “unnamed”

make a variable called self.hitPoints, give it the value 0

make a variable called self.hitChance, give it the value 0

make a variable called self.maxDamage, give it the value 0

make a variable called self.armor, give it the value 0

define function called propertyMaker, given the parameters self, value, trait

make a getter property

define a function called “trait”

return the name of that trait

make a setter property

define a function called trait with the parameters “self” and “value”

give the trait the argument “value”

define a function called testInt with the parameters “character” “traitName” “trait” “minimum” and “maximum”

#each integer should actually be an integer and not some other type

#each integer should be nonnegative

Give keepGoing True

While keepGoing is equal to True

The variable traitResponse gets the user’s response to “What would you like {character}’s {traitName} to be?”

If the traitResponse is an integer

If the argument is bigger than “maximum”

If the argument is smaller than “minimum”

Assign traitResponse to character’s “trait”

Return the character’s trait value

Give keepGoing False

Otherwise print “Sorry, that’s larger than the excepted maximum: {maximum}. Please try again”

Otherwise print “Sorry, that’s smaller than the excepted minimum: {minimum}. Please try again”

Otherwise print “Sorry, that’s not an integer, please try again”

define a function called printStats() with the parameters traitName and trait

#this function will be called with each trait passed in individually so that it can still handle if new traits are created in other versions of the game

print the character’s traits and the corresponding value

define a function called statValueGetter() with the parameter playerStat

Generate a random number between 0 and the value assigned to that stat

Return the number generated as the variable statValue

Define a function called

define a function called doFight() with the parameters personOne and personTwo

print that the personOne hit personTwo

the variable damageDealt gets the result of the function statValueGetter() with the parameter maxDamage passed

print how much personOne hit them for using {statValue}

the variable armorBlocked gets the result of the function statValueGetter() with the parameter armor passed

print how much personTwo’2 armor absorbed

if armorBlocked is more than or equal to damageDealt

health stays the same

if armorBlocked is less than damageDealt

health gets the original health minus damageDealth plus armorBlocked

return health

Build program called combat.py

import tbc module

Define the main function

Give the variable “c” Character() in order to be able to change the values inside the class

The variable playerName gets the user’s response to “What would you like to name your character?”

Player’s Health gets return from Call testInt with the parameters playerName, health, hitPoints, 0, 100

Player’s hitChance gets return from Call testInt with the parameters playerName, success rate, hitChance, 0, 100

Player’s maxDamage gets return from Call testInt with the parameters playerName, strength, maxDamage, 0, 100

Player’s armo gets return from Call testInt with the parameters playerName, armor absorption, armor, 0, 100

Monster’s health gets return from Call testInt with the parameters the monster, health, hitPoints, 0, 100

Monster’s hitChance gets return from Call testInt with the parameters the monster, success rate, hitChance, 0, 100

Monster’s maxDamage gets return from Call testInt with the parameters the monster, strength, maxDamage, 0, 100

Monster’s armo gets return from Call testInt with the parameters the monster, armor absorption, armor, 0, 100

Give the variable player tbc.Character() with the parameters of all of their stats passed in

Give the variable monster tbc.Character() with the parameters of all its stats passed in

Use a for loop and printStats() to let the user know what the current stats are

Set keepGoing equal to True and start a while loop

use doFight() to run first half round of the fight with the parameters player, monster

use doFight() to run second half of the fight with the parameters monster, player

the variable nextRound gets the response to “Move on to next round? (Y/N)”

if nextRound made uppercase is equal to “N”

give keepGoing False

else if nextRound made uppercase is equal to “Y”

call doFight()

otherwise nextRound

print “Sorry, that’s not a valid response, please try again”

if player’s health is equal to or less than 0

if monster’s health is equal to or less than 0

print “It’s a tie! No one wins”

else

print “Sorry {player} you lose!

Else if monster’s health is equal to or less than 0

Print “Good job {player}, you win!”

Call main using \_\_name\_\_ == “\_\_main\_\_”