TBC

Assign random (import random)

Assign class Character (object)

Define init as (self, name = "default name", hitPoints = 6, hitChance = 8, maxDamage = 8, armor = 1)

Assign super init

Assign self.name = name

Assign self hitPoints = hitPoints

Assign self hitChance = hitChance

Assign self maxDamage = maxDamage

Assign self armor = armor

Assign property

Define name as (self)

Return self name

Assign name setter

Define name as (self, value):

Return self name to get value

Assign property

Define hitPoints(self):

Return self hitPoints

Assign hitPoints setter

Define hitPoints as (self, value)

Return Self hitPoints to get value

Assign property

Define hitChance(self):

return self hitChance

Assign hitChance setter

Define hitChance(self, value)

Return Self hitChance and it get value

Assign property

Define maxDamage(self):

return self maxDamage

Assign maxDamage setter

def maxDamage(self, value):

Return self maxDamage to get value

Assign property

Define armor(self):

return self armor

Assign armor.setter

Define armor(self, value):

Return self maxDamage to get value

define testInt(self, value, min = 0, max = 100, default = 0):

take in values

checks to see if it is an int between

min and max. If it is not a legal value

set it to default

out get default

if type(value) == int:

if value >= min:

if value <= max:

out = value

else:

print("Too large")

else:

print("Too small")

else:

print("Must be an int")

return out

define printStats(self):

Then print

{self.name}

hit Points: {self hitPoints}

hit Chance: {self hitChance}

max damage: {self maxDamage}

armor: {self armor}

define hit(self, Runner):

hitRand get random.randint(0, 100)

dmgNum get random.randint(1, self maxDamage)

if hitRand <= self hitChance:

Print: self.name hits Runner.name

for dmgNum points of damage

Then Runner.name armor absorbs Runner.armor points of damage

if dmgNum > Runner.armor:

dealtDmg = dmgNum - Runner.armor

Runner.hitPoints -= dealtDmg

else:

Print: self.name missed

Define fight as (Chaser, Runner):

Then

keepGoing gets True

while keepGoing:

Define hit to get (Chaser, Runner)

Define hit to also get(Runner, Chaser)

Print Chaser name and RunnerhitPoints to HP

Print Runner name Chaser hitPoints to HP

if Chaser.hitPoints > 0:

if Runner.hitPoints > 0:

userChoice = input("Press ENTER for another round or Q to quit: ")

if userChoice == "Q":

keepGoing = False

else:

print(f"{Chaser.name} wins")

keepGoing = False

else:

print(f"{Runner.name} wins")

keepGoing = False

Define main():

c = Character()

printStats(c)

Assign

if \_\_name\_\_ == "\_\_main\_\_":

main()

Combat

import Marvellous\_TBC

define main

Assign runner to get Marvellous\_TBC.Character()

Assign runner name to be "Runner"

Assign runner hitPoints to 6

Assign runner hitChance t0 8

Assign runner maxDamage to 8

Assign runner armor to 1

Assign chaser to get Marvellous\_TBC.Character

Assign name to, hitPoints to 7, Assign hitChance to 8, Assign maxDamage to 7 Assign armor to 1

Set Marvellous\_TBC.printStats to get (chaser)

Set Marvellous\_TBC.printStats to get (runner)

Marvellous\_TBC.fight gets(runner, chaser)

Assign

if \_\_name\_\_ == "\_\_main\_\_":

main()