Pseudo code:

main() code that we are basing our codes on:

```
import tbc

def main():
    hero = tbc.Character()
    hero.name = "Student"
    hero.hitPoints = 15
    hero.hitChance = 55
    hero.maxDamage = 5
    hero.armor = 2

    monster = tbc.Character("Professor", 30, 35, 6, 0)

    hero.printStats()
    monster.printStats()

    game = tbc.Fight(hero, monster)
    game.fightscene()

if __name__ == "__main__":
    main()
```

TBC module:

Import random

Make a new class called **Character**
Initialize the values of self, **name, hitPoints, hitChance, maxDamage,** and **armor** to zero or other appropriate value.

Assign self.__xxx for each value

Create a property and a setter for each value

Def **printStats(self)**
Print each value

Make a new class called **Fight**
Initialize the value of self, hero, and monster and assign **self.hero** to hero, and **self.monster** to monster.

Def attack (self, attacker, defender)
Create a variable called **chance** and assign it a random integer between 1~100 using random.randint

Make a condition where when **chance** is within the attacker's hitChance, it will create a variable called **damage,** which will be a random number between 1 and the attacker's maxDamage. If **chance** is not within the attacker's hitChance, print that the attacker missed the shot. After assigning **damage,** create a variable called **absorbed** where you store absorbed damage such as **defender.armor.** Make the program not absorb more than zero by using min function, which choses smaller integer between defender's armor and the actuall damage so it won't absorb more than the armor's value. Then, create an another variable called **finalDamage** where we adding the final amount of damage that are gave to the defenders.Subtract finalDamage from defender's hit point.

Create a function called **fightscene(self),** where you call the **attack module** for both hero and the monster until one of their hit point reaches zero.Use while keepGoing to do this.