P5HW\_UsingAI - Demo Script

Intro  
Hi, my name is Kimberly Simmons, and this is my program for assignment P5HW\_UsingAI.  
The purpose of this game is to let the user create characters, view their stats, and have them battle each other.  
I will now explain each function in my code and show you the game running.  
  
create\_character() Function  
First is the function called create\_character()  
- This function has no parameters.  
- It asks the user to enter a character name, health (as an integer), attack points (integer), and defense points (integer).  
- It stores all of that information in a dictionary and then returns that dictionary.  
- Because it returns the character data, it is a value-returning function.  
- The return type is a dictionary, because we need to keep track of multiple character stats together.  
- This value is stored and added to a list so we can manage multiple characters.  
  
display\_character(character) Function  
The second function is display\_character(character)  
- It takes one parameter, which is the character dictionary.  
- It goes through each key-value pair in the dictionary and prints them out in a clean format.  
- This function does not return anything, so it’s a non-value returning function.  
- Its job is just to show the data to the player.  
  
attack(attacker, defender) Function  
Next is the attack(attacker, defender) function.  
- It has two parameters, both are dictionaries that represent two characters.  
- The function calculates damage by subtracting the defender’s defense from the attacker’s attack.  
- If the result is negative, it becomes zero so no healing happens.  
- Then it reduces the defender's health by that damage amount.  
- If the defender’s health drops to zero or below, the function prints that the character has been defeated.  
- This is a non-value returning function, because it updates the dictionary in place and just prints the results.  
  
main() Function  
The main() function is the heart of the program.  
- It doesn't take any parameters.  
- It contains the game loop and controls the menu system:  
 1. Create characters  
 2. Display characters  
 3. Attack between characters  
 4. Exit the game  
- When you create a new character, it calls create\_character() and stores that in a list.  
- When you choose to display them, it loops through each one and shows their stats.  
- When you battle, it lets you pick an attacker and a defender using numbers, then uses the attack() function to simulate the fight.  
- This function is non-value returning, and everything is coordinated from here.  
  
Now I’ll play through the game once  
1. I’ll create two characters:  
 - One named “Hero” with 100 health, 30 attack, and 10 defense.  
 - One named “Villain” with 100 health, 25 attack, and 5 defense.  
2. I’ll display both characters to make sure they’re saved correctly.  
3. I’ll choose Hero to attack Villain.  
 - The damage is 30 - 5 = 25, so Villain should lose 25 health.  
4. I’ll display the characters again to see Villain’s health is now 75.  
5. I’ll exit the game.  
  
Wrap-Up  
That completes my demo.  
I’ve shown all required functions, explained their parameters and return types, and walked through the game logic.  
Thank you!