Homework #4
Text Adventure Multifile Project

Overview

For this homework assignment you are refactoring your homework #2 assignment from a simple parallel array system in a single file program to a multifile Dev-C++ project with structs, enumerations, typedefs, dynamic memory, and file input and output. This will exercise your ability to take already existing code and adapt it to new uses and conditions, strengthen your concepts of data structures and give you the opportunity to make a complete program with many different parts.

Requirements:

Your program should be refactored to meet the following requirements:

- All parallel array structures are converted to use structs and arrays of structs.
- All uses of hard coded indicators and flags will be converted to use Enums.
- Where convenient, typedefs will be used to convey the use of particular data.
- All functions will be grouped according to their purpose in separate header and source files. This means using more than one file in your project, with as many as necessary to clearly define and organize your code.
- All code should be properly commented, variables and functions named in a logical manner, and well structured (proper indentation.)

Additionally, your program should add the following features:

- A dynamic inventory which allows any number of items to be added to the player (proper use of dynamic memory management) … this should remove items correctly as well.
- A save system which saves the status of the player into a file. This does not have to save every single piece of information, but at a minimum it must save the name of the player, and the health of the player.
- The ability to load the game from a previous play through by opening and reading the save file.

Rubric

Program Compiles ….	**10 pts**
Structs used correctly according to requirements…	**10 pts**
Enums used correctly according to requirements…	**10 pts**
Typedefs used appropriately…	**10 pts**
Mulitfile project correctly implemented …	**20 pts**
Functions grouped in a logical manner …	**10 pts**
Dynamic inventory implemented correctly …	**10 pts**
Program can save progress to file correctly …	**10 pts**
Program can load progress from file correctly …	**10 pts**
Total …	**100 pts**