For this homework you will be creating the structure for a Role Playing Game (RPG). You will be adding a class called RPGCharacter, a class called CombatCharacter, one called a NonCombatCharacter, at least two combat character classes like: Barbarian, Bard, Cleric, Druid, Fighter, Monk, Paladin, Ranger, Rogue, Sorcerer, Wizard, etc., and finally, at least two non-combat character classes like: ShopOwner, Townie, etc.

The RPGCharacter class will be designed to represent a type of character in a Role Playing Game.  Some of you are probably already familiar with the idea of a RPG but even if you are not don’t worry – I could have just as easily asked for a Professor class or a Veterinarian class, and Athlete class, a Sensei (a karate instructor) or really just about ANY class – they are ALL done the same way.  All you will need to know is what data and methods you should create for your classes.  Also, you should note that each combat character type could have a different range in their abilities.  For instance, a fighter might have a Strength level between 1 and 20, whereas, a Barbarian’s Strength range might be from 5 to 25.  Consider the following **short** list of attributes / properties for a Fighter:

1. name................... The name of the fighter
2. level..................... An int from 1-20. Represents the fighter’s rank or level
3. experience........... An int that represents the accumulation of points obtained during battles and other things
4. strength............... An int from 1-20. Represents the fighter’s strength
5. dexterity.............. An int from 1-20. Represents the fighter’s agility
6. charisma.............. An int from 1-20. Represents the fighter’s personality. How good they are with people/animals
7. battlesWon.......... An int that represents the number of battles the fighter has won.
8. speed.................. An int from 0-10. Represents how fast the fighter is moving
9. life....................... An int from 0-100. Represents the fighter’s life points. 0 = dead. 100 = full life.

Of course there are more attributes that could and should be included in a Fighter class.  In addition to the attributes / properties of the fighter we should include some things the fighter can do.  These are called methods.  Consider the following **short** list of methods for a fighter (you must include the **fight**method with some basic fighting code if the character is a CombatCharacter):

1. **run()**...................... A method that will indicate the fighter is running. It might set a speed property higher
2. **walk()**.................... A method that would probably set the fighter’s speed lower than running but higher than 1
3. **stop()**.................... A method that will set the fighter’s speed to 0.
4. **talk()**..................... A method that simulates talking to someone/something and uses the fighter’s charisma value
5. **fight()................... A method that controls a battle. Attributes (Level, Strength, Dexterity, etc.) will be used.**
6. **quickGenerate().. A method that will randomly set the values of the attributes of the fighter**

There are many other attributes and things a character could do.  Your job at this point is to create the base classes so that you force child classes to include various methods.  Your base class can contain general properties and methods (e.g., Name, setName(), getName()) You should then create at least 2 combat and 2 non-combat child classes. For **some**of the methods you should include basic code. Optionally, you can add code to many of the methods.

Each of the characters **must** have a quickGenerate() method that is called from the constructor of that class and will randomly set all of the attributes of that character taking into account that each character has a different range for each of the obvious attributes (hint: this should be an abstract method in the RPGCharacter class) - basically, this would make for a quicker game because each character will start the game with random data rather than having to play for hours to gain experience, gold, etc. It would be the same thing as starting a game of Monopoly and dealing out all the properties in order to save hours of playing.

Each of the combat characters **must** have a fight(RPGCharacter character) method that is used to battle. When called, the fight() method will determine which character (this or character) gets to hit the other. It will then determine if contact was actually met. Finally, it will subtract the number of points from the “losing” RPGCharacter’s life.  Rule: if a combat character fights a non-combat character the combat character always wins.