

BACK-END

Description	Score (Total:70)
Understanding of requirements and implementation	30
Use of Object-Oriented Programming (OOP) (Class, Inheritance, Encapsulation)	20
Code quality (Variables, Methods, Code usages, Code flow)	20

In this question, the following skills will be assessed:

- Object Oriented Programming (OOP)
- C# and .NET
- Requirement Understanding
- Class Design
- Code Quality
- Troubleshooting skills

Tool can be used:

- Visual Studio

Below are the requirements for the console:

1. You as a **human player** will face a **computer AI** in the **mortal game of rock-paper-scissors**.
2. At the start of the game, **system will ask the player to enter their name**.
3. Both the player and the computer will have **3 health points/hit points each**.
4. **During each turn, the player will first provide an input to choose from one of three available actions.**
 - For example; 'q' for rock, 'w' for paper and 'e' for scissors. (Note: You can also decide to define the inputs as you see fit in your implementation.) After the player input, the computer will randomly pick from the three available choices as well.
5. The result of the turn will then be automatically decided based on their choices:
 - **Rock will beat the scissors, paper will beat the rock, and scissors beats the paper.**
 - **If the player beats the computer during the turn, the computer will lose 1 health. If the computer won the turn, the player will lose 1 health.**
 - **If both the player and the computer picked the same choice, the turn will become a tie.** Neither of them will lose health and the game continues to next turn.
6. At the end of every turn, the game has to display the result as output on screen as follows.
 - The choices both the player and the computer made
 - Whether the player won or lost the current turn
 - And the remaining health for both player and computer
7. The game goes on until either the player or the computer lost all 3 health points and dies.
Additional requirements for bonus. (Note: Analyze these requirements carefully before you proceed.)
8. There will be a **fourth input option** only for the player, to **use a magical potion** in their possession. The potion can restore 1 health point if the player choose to drink it during the input phase. If the player chose to drink the magical health potion, the computer will not take any actions and skips

the current turn.

9. The same magic potion can save the player from dying.

If the player is down to 1 health point and the computer makes a move that would normally defeat the player, there's a 50% chance the potion will be used automatically to save them. But this will only happen if the potion hasn't been used yet in the game.

It works like the earlier rule, but this time, it only happens by luck at the end of the turn—just before the player's health drops to 0.