Initial Design

I will be modelling my system on two player interfaces. This will be a two-player computer game where the player gets to choose between four types and fight to win by getting opponent's health to zero. Players has some attributes like Health, Armour, Xp and Powerups, which they use in the process of defeating or defending the opponent.

Memory allocation:

The game will split into different objects that will perform specific tasks. An example of which can be creating four different types of character objects to start from which is asked at the initial stage of the game.

Memory management:

To make that there is no memory leak in my program, every boundary case will be tested for any anomalies. Also, at points where dynamic memory is generated, it will be made sure to free the memory created.

Array:

The stats of the player and individual characteristics will be stored in array and most probably in form of stack memory as it is already decided. Usernames of the players will be stored in form of array.

Strings:

The players will be printed in the terminal using randomly generated string of characters. The usernames are also stored as string inputs. After every fire or attack, a custom-made output is displayed which is made from the attack used and is stored as string input.

User Input:

The program will take a variety of user inputs. When the game starts, the screen will present with game information and will ask for number of rounds to play (user input in integer type). Then the screen will ask for string user input for two players in the game. Finally, during the game the users can hit "g" (gunshot), "s" (sword), "c" (cannonball), and many more can come up while developing the game.

Inheritance:

All four-character type will inherit from the main person class. Different attacking types will inherit from the main person class to modify the attack chosen by the player and update the status.

Polymorphism:

Testing

Each and every step in my program will be tested till its standard are reached. After every class made, it will be tested by a testing file to check every possible cases or boundary cases to qualify it to next level.