From last week’s code, I made the necessary changes to fix what didn’t follow the directions and cost me points. I added the character classes to their own package and added an abstract method along with an individual toString() method in each individual type of character. After that I changed my fight simulation code to work on any two given indexes in an arrayList so that I could randomly select tow fighters in the Arena ArrayList and it would still operate. I let the user select how many fighters they would like in the Arena and then simulate fights between two randomly chosen fighters from the arena. After the winner is determined I created a heal method to bring them back to full health in order to let them continue fighting without an unfair advantage for the new challenger. When a fighter loses they are removed from the Arena so I must regenerate the random selection using new indexes because some or possibly even all fighters have new indexes in the list (if the first fighter, at index 0, in the list is the one that lost all of the fighters will be moved up, or down depending on how you look at it, by one index). **My code will work with as many fighters as can be stored in an int datatype meeting the 4 fighter requirement.** This is due to the user’s input for number of fighters being stored as an int and the number of fights being stored as an int. There will always be one less fight than the number of fighters because all but one must be eliminated to find a winner. Graphical user interface, text

Description automatically generatedText

Description automatically generated with low confidence