1. The user runs the software program, which warmly welcomes him to “KnightFight”.  
2. The user is asked for the name of his Knight, which he enters with the keyboard.   
3. The user is asked what weapon his Knight is using, which he enters with the keyboard.  
4. The user is asked if he wishes to auto-generate his opponent.   
(a) If the user responds “yes”, the software will auto-generate his opponent with randomzied attributes.   
(b) If the user responds “no”, the user is able to define the name and weapon type of his opponent.  
5. The software will now give each Knight a random number of hitpoints.  
6. The software now displays a brief summary of each Knight, showing his name, weapon, and hitpoints.  
7. The software asks the user to press any key when they are ready to begin OR takes a y/n answer to start or exit.   
8. Upon recieving the key, the software randomly chooses which Knight shall begin combat.   
9. Each Knight attacks in turn with their chosen weapon, which deals a random amount of damage based upon the weapon type.   
10. The damage dealt is subtracted from the hitpoints of the opposing Knight.   
11. A brief summary of each round is displayed to the user.   
12. Once a knight's hitpoints are reduced to zero or less, the fight ends with the Knight with hitpoints reminaing declared the winner.   
  
Welcome to KnightFight!  
Enter the name of your Knight: Bob  
  
Now select your weapon! (Choose number)  
1) Long Sword  
2) Battle Axe  
3) Spear  
4) Warhammer  
Your choice my liege? : 3  
Would you like to auto generate your opponent? (y|n)  
  
1) Two constructors for the Knight class, one for an auto-generated Knight and another for the user-defined Knight.   
2) A switch statement must be used to determine weapon selected. Those using Java 7 may use a String to determine their case selected.   
3) The Java Random class must be used where any random element is required.   
4) You may use a simple GUI or the console. Do not let yourself get bogged down in GUI building, this assignment is designed to test your usage of the logic components of the Java language.   
5) The Knight class will have a toString() method to display all attributes, a fight() method to determine damage, and setters/getters for all instance variables.   
6) Deliverables are the Knight class and the KnightDriver class.   
7) Allow the user to restart the game without exiting the program.   
8) Give each Knight an armor type and alter the damage amounts based upon weapon type vs armor type. (ie blades do more against leather and less against metal) Hints: A semaphore in a while statement could be useful to alternate actions...