GitHub Location: <https://github.com/GeorgeKoralski007/Refactoring>

**Refactoring Documentation**

1. Delete unused code: Comparable and compareTo from DungeonCharacter class(George)
2. Specified the variables that are being passed in the constructors of the Monster classes(I did this because as is the numbers that are passed into the super are very arbitrary on what they actually are. (Matthew)
3. Made the Monster Class Protected as well as Made heal(), chanceToHeal, minHeal, and maxHeal to private from protected. Also made all of the methods in DungeonCharacter protected(matthew)
4. (not sure if needed)made getHealPoints() and getCanHeal() in the monster class.
5. (Still not sure if needed) made a private getDamage() to further breakdown the code in the dungeonCharacter class
6. playAgain() validates input to accept only ‘y’ or ‘n’ (George)
7. chooseHero() validates input for range 1 to 3 (George)
8. Create a factory class DungeonCharacterFactory to create Hero and Monster. Decouple user input from object creation. (George)
9. battle() change the while to do-while loop because the first entry is always a true condition.(George)
10. battle() rename ‘pause’ to ‘choice’ and validates user input. Improve prompt message (George)
11. Dungeon most public static methods converted to private non static. Adding public play(). (George)
12. addHitPoints and subtractHitPoints make protected (George)
13. Changed how the battleChoices() method works. It now uses a helper method attackMenu(String, DungeonCharacter) in the hero class to run the battleChoices() menu. (Andrew)
14. Added static fields for the supercall values in hero subclasses in order to decrease ambiguity and improve readability on the constructor’s super call. (Andrew)
15. Changed visibility of defend() to private as it’s a helper method that doesn’t need to be accessed outside of Hero.java’s subtractHitPoints() method. (Andrew)
16. Added comments to special attack methods of each hero. Also made the method private. (George)
17. Make Hero.battleChoices abstract and move the code to initializeTurns (George)

Bonus points:

1. Add specialSkill to Monster (George). Hero already has special skills so defining the abstract specialSkill to DungeonCharacter.
2. Added King Hero with special attack (George)