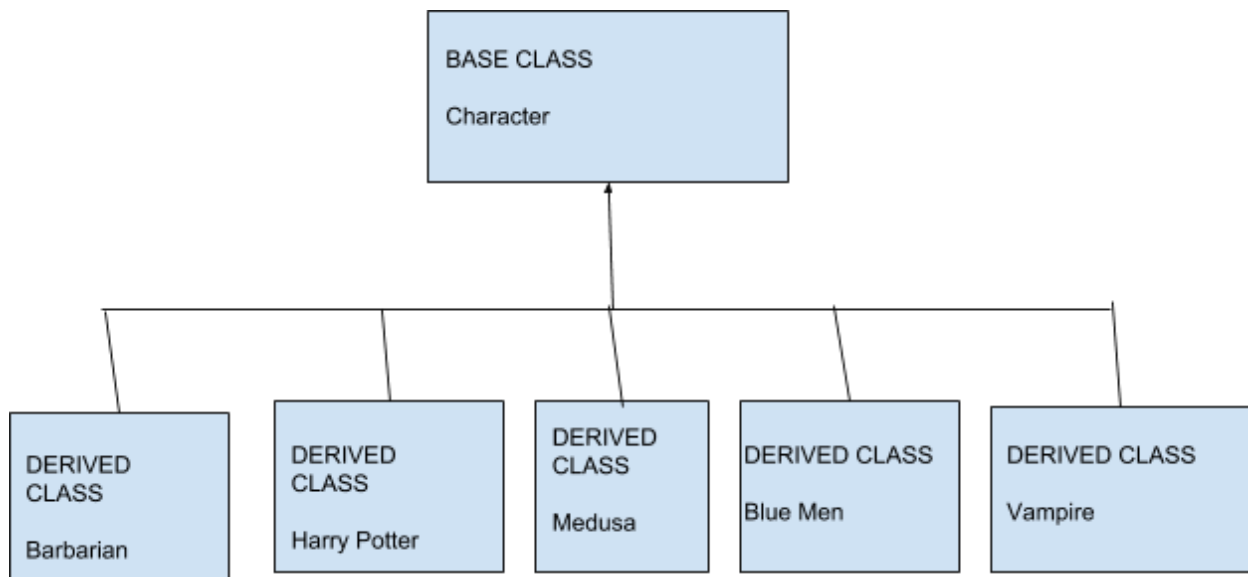**DESIGN:**

To start, I wrote down the base class requirements, and then the requirements for the derived classes. All in list form. I didn't write down the specifications for each characters, just the assumed variables(strength, armor, etc). After that, I listed possible menu functions (display_results, play/exit), and then after that I wrote down a brief synopsis of the game play, which I wanted to be part of my Game class I would create. I assumed that most likely, my game class would include 2 functions, menu and play game. Play game should include 1. Combat occuring 2. Player attacks 3. Damage calculated/subtracted. After this, I decided to rewatch the lecture on polymorphism while writing my base class(character), and its virtual functions of attack/defense, and then what the play game function will take as input to point to the characters. So, I basically created a rough skeleton of my game and left out the details. I made some mistakes along the way with switch/if statements and their scope (with character creation), not using dynamic memory at first(led to segmentation faults), and assigning pointers incorrectly.

REFLECTION:

If I could do this project over, I would make my menu, character selection, and combat more succinct. I know there's a better way of implementing the character allocation, but having to work with creating a pointer to a character object, then assigning a derived dynamically allocated object to that character got a bit cumbersome and I was just really nervous to incorrectly declare them, so I kept the menu very repetitive. I would definitely create a class just for menu functions, and one for input validation if I could do it over, and would like to make these for my future projects; sometimes it just complicates things for me as a new programmer, even though theoretically if I were a pro at pointers or dynamic memory, for example, I'd be more confident in doing so.

CLASS HIERARCHY DIAGRAM:

TEST CASES:

Test Case	Input Values	Driver Functions	Expected Outcomes	Observed Outcomes
Vampire Barbarian	1 2	my_menu	Vampire beat Barbarian	Round 1: Character: Barbarian Strength: 12 Armor: 0 Attack: 9 Character: Vampire Strength: 12 Armor: 1 Defense: 2 Vampire beat Barbarian Round 2: Character: Vampire Strength: 12 Armor: 1 Attack: 8 Character: Barbarian Strength: 10 Armor: 0 Defense: 6 Vampire beat Barbarian Round 3: Character: Barbarian Strength: 10 Armor: 0 Attack: 6 Character: Vampire Strength: 10 Armor: 1 Defense: 3 Vampire beat Barbarian Round 4: Character: Vampire

				Strength: 10 Armor: 1 Attack: 3 Character: Barbarian Strength: 10 Armor: 0 Defense: 8 Vampire beat Barbarian Round 5: Character: Barbarian Strength: 10 Armor: 0 Attack: 6 Character: Vampire Strength: 6 Armor: 1 Defense: 1 Vampire beat Barbarian Round 6: Character: Vampire Strength: 6 Armor: 1 Attack: 5 Character: Barbarian Strength: 10 Armor: 0 Defense: 9 Vampire beat Barbarian Round 7: Character: Barbarian Strength: 10 Armor: 0 Attack: 8 Character: Vampire Strength: 2 Armor: 1 Defense: 3

				Vampire beat Barbarian Round 8: Character: Vampire Strength: 2 Armor: 1 Attack: 3 Character: Barbarian Strength: 10 Armor: 0 Defense: 4 Vampire beat Barbarian Round 9: Character: Barbarian Strength: 10 Armor: 0 Attack: 8 Character: Vampire Strength: 0 Armor: 1 Defense: 1 Barbarian beat Vampire
Vampire Blue Men	1 3	my_menu	Vampire win	Round 1: Character: Vampire Strength: 18 Armor: 1 Attack: 6 Character: Blue Men Strength: 12 Armor: 3 Defense: 11 Vampire beat Blue Men Round 2: Character: Blue Men Strength: 12

				Armor: 3
				Attack: 5
				Character: Vampire
				Strength: 18
				Armor: 1
				Defense: 4
				Vampire beat Blue Men
				Round 3:
				Character: Vampire
				Strength: 18
				Armor: 1
				Attack: 10
				Character: Blue Men
				Strength: 12
				Armor: 3
				Defense: 11
				Vampire beat Blue Men
				Round 4:
				Character: Blue Men
				Strength: 12
				Armor: 3
				Attack: 14
				Character: Vampire
				Strength: 11
				Armor: 1
				Defense: 6
				Vampire beat Blue Men
				Round 5:
				Character: Vampire
				Strength: 11
				Armor: 1
				Attack: 6
				Character: Blue Men
				Strength: 12
				Armor: 3
				Defense: 10
				Vampire beat

				Blue Men
				Round 6: Character: Blue Men Strength: 12 Armor: 3 Attack: 11 Character: Vampire Strength: 4 Armor: 1 Defense: 3 Vampire beat Blue Men Round 7: Character: Vampire Strength: 4 Armor: 1 Attack: 5 Character: Blue Men Strength: 12 Armor: 3 Defense: 10 Vampire beat Blue Men Round 8: Character: Blue Men Strength: 12 Armor: 3 Attack: 13 Character: Vampire Strength: 0 Armor: 1 Defense: 3 Blue Men beat Vampire
Barbarian Blue Men	2 3	my_menu	Blue men win	Round 1: Character: Blue Men Strength: 12 Armor: 3

				Attack: 15 Character: Barbarian Strength: 5 Armor: 0 Defense: 8 Barbarian beat Blue Men Round 2: Character: Barbarian Strength: 5 Armor: 0 Attack: 7 Character: Blue Men Strength: 12 Armor: 3 Defense: 7 Barbarian beat Blue Men Round 3: Character: Blue Men Strength: 12 Armor: 3 Attack: 11 Character: Barbarian Strength: 0 Armor: 0 Defense: 5 Blue Men beat Barbarian
Medusa Harry Potter	4 5	my_menu	Harry potter win	Round 1: Character: Harry Potter Strength: 10 Armor: 0 Attack: 6 Character: Medusa Strength: 6 Armor: 3 Defense: 1

				Medusa beat Harry Potter
				Round 2:
				Character: Medusa
				Strength: 6
				Armor: 3
				Attack: 10
				Character: Harry Potter
				Strength: 9
				Armor: 0
				Defense: 9
				Medusa beat Harry Potter
				Round 3:
				Character: Harry Potter
				Strength: 9
				Armor: 0
				Attack: 8
				Character: Medusa
				Strength: 4
				Armor: 3
				Defense: 3
				Medusa beat Harry Potter
				Round 4:
				Character: Medusa
				Strength: 4
				Armor: 3
				Attack: 6
				Character: Harry Potter
				Strength: 7
				Armor: 0
				Defense: 4
				Medusa beat Harry Potter
				Round 5:
				Character: Harry Potter
				Strength: 7
				Armor: 0
				Attack: 8

				Character: Medusa Strength: 0 Armor: 3 Defense: 1 Harry Potter beat Medusa
Exit	2	my_menu_2	Say goodbye, exit program	Say goodbye, exit program
Play Again	1	my_menu_2	Go back to my_menu	Went back to my_menu
String input	s	my_menu	Say invalid, ask for number	"Invalid. Enter integer between 1-5"
String input	t	me_menu_2	Invalid, enter 1 or 2	"Invalid. Enter either 1 or 2"