William Dang
CS 162
Project 3 Design/Reflection

Design/Reflection:

Similarly to the previous project, polymorphism was used in this project. Therefore, creating a base Character class and its subclasses was relatively straightforward. I started out by figuring out which variables each character type shared in common, which ended up being character type, attack, defense, armor, and strength points. I also made the status of win/defeat a common variable as well. Next, I planned to create getters for each one of these variables in the base abstract class (Character) and created the header files for all the character types (Barbarian, BlueMen, HarryPotter, Medusa, Vampire) which were very similar to the base class header file. Next, I designed the implementation files for each character type. I made sure to follow the specifications for the die/dice rolls seen in the project requirements. For the Barbarian class specifically, I wrote the formulas to return a randomized number on a six sided die for attack and defense (since the same, "2d6" was specified for both attributes). To represent that two dice were necessary, I added the original formula's result to the same formula again to produce another randomized integer (from 1-6) to add. Because attack and defense points are generated by these rolls, I made sure to initialize these variables within the default constructor. Getters were made for each attribute, including getters for the rolls for attack and defense. This process was very similar to what I did for the other character types, making sure to base their attributes and attack/defense rolls off of program specifications.

My initial plan for the main function was generally fine, but I didn't want to think about/account for the special abilities until I solved an issue with my function to get defense after a round of attacking. Using my test function, I noticed that my get function for defense needed to be changed because it wasn't correctly calculating damage from attack for each round. Instead of doing all of the calculations within one function in main as I had initially planned, I made a new defend function which correctly accounted for damage in the round by taking attack parameter and creating a damage variable to work with. After correcting this error in calculation during these rounds, my project was able to compile. As stated in the project requirements, certain classes produced much higher chances to win based on special abilities or certain attributes (which made it easier to debug and test). After testing each case, the actual outcome matched the expected outcomes I had. For instance, the HarryPotter character won many times because of his special ability that reset his health to 20 after his first death. This usually allowed Harry Potter to outlive his opponents by later rounds because he essentially had 30 strength points. However, whenever damage was a negative value, I did run into an error whenever damage was subtracted from strength points. It caused strength points to increase rather than decrease, which was not the intended effect. Therefore, I updated my input validation so that when damage was a negative value, strength points would not change. Other than that, everything met my expectations after testing.

Character.hpp (base class, subclasses are very similar)

Public member variables:

int attack_Point;
int defense_Point;
int armor_Point;
int strength_Point;
bool defeat;

Character.cpp functions (not using a .cpp file, but subclass source files include these functions)

Constructor (initialize with armor and strength according to project description, initialize attack and defense points to 0 because these are determined by rolled die/dice, initialize defeat condition

Destructor

Attack_roll function

Defense_roll function

Getters (character type,k strength, armor)

Bool defeated condition

Attack function

Defend function (should include damage variable for calculation)

In the subclass source files, make sure Harry Potter has special ability (set strength points to 20) when He dies.

Medusa "glare" gives her instant win, so maybe set damage to a high value to win every time the special ability occurs (every time 12 is rolled).

For Vampire special, set charm effect to occur with 50% chance using rand()%2+1.

For the BlueMen class, implement the "Mob" special by using one less dice with every 4 strength points lost.

Main.cpp

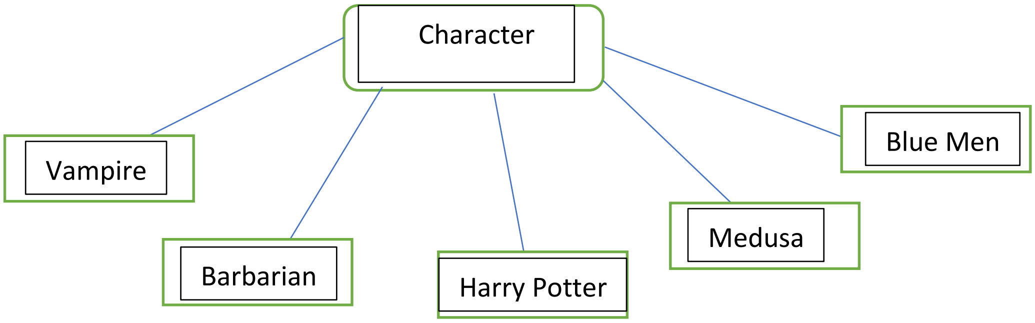
Menu prompts and options displayed

Switch case statements for player_1 and player_2 to select a character from 5 options

Display results at end of each round (defense points, armor, damage received, strength points left)

Loop for user to play again if desired/exit

Class Hierarchy Diagram: (all lines here represent "is-a" relationships)



Test Plans/Results:

Test Case	Expected Output	Actual Output
Player 1 chooses Harry Potter Player 2 chooses Medusa (attempted to select for every character type for each player)	Print prompt showing character type selected for the player	Player 1: printed prompt showing that player 1 selected Harry Potter. Player 2: printed prompt showing that player 2 selected Medusa. Printed prompt as expected for other character types, showing character type selected for the player.
Player 1's character's strength points reach zero.	Print victory message for Player 2	Printed victory message for Player 2
Player 2's character's strength points reach zero.	Print victory message for Player 1	Printed victory message for Player 1
Harry Potter character reaches 0 or below strength points on his first life.	Print message that he revived and set strength points back to 20	Printed message that he revived and set strength points back to 20.
Harry Potter character reaches 0 or below strength points after having used "hogwarts" once.	Harry Potter dies and other player wins.	Harry Potter died and other player won.
Medusa "glare"	Medusa wins (because damage set to 100) unless against Harry Potter who revives against her special on his first life. "Glare" kills Harry Potter on his second life.	Medusa won against other characters on their first life, unless against Harry Potter who revived against her special on his first life. "Glare" kills Harry Potter on his second life.
Blue Men "Mob" special	Prompt and set: 3 defense die when strength points 9 to 12. 2 defense die when strength point = 5 to 8 1 defense dice when strength point = 0 to 4.	Prompted and set: 3 defense die when strength points = 9 to 12. 2 defense die when strength point = 5 to 8 1 defense dice when strength point = 0 to 4.
Vampire "Charm"	No change in vampire's own strength points when special occurs	No change in vampire's own strength points when special occurred
User input to play game again	Return to menu prompt to pick Characters again for each player.	Returned to menu prompt to pick Characters again for each player.
User input to exit game	Print prompt for exiting and game exits	Printed prompt for exiting and game exited
Player's character receives negative value for damage	Strength points unaffected rather than increased (due to validation written for this)	Strength points unaffected (due to validation written for this)