Project 4

Design:

Creature class

- Protected data members for name(type), armor, and strength points
- Pure virtual destructor and 2 getter member function
- Virtual functions for attack, defense, and pure virtual function for damage
- Enumeration declaration of Special Abilities
- Virtual function for recovering strength points and change name

Vampire class

- Inherits from Creature class
- Private data members for attack and defense values
- Constructor that assigns unique values to type, armor, and strength points and seeds a random number generator
- Override defense and damage function
- If defense roll was greater than 3, then Vampire uses Charm to reduce upcoming damage to 0.

Barbarian class

- Inherits from Creature class
- Private data members for attack and defense values
- Constructor that assigns unique values to type, armor, and strength points and seeds a random number generator

Medusa class

- Inherits from Creature class
- Private data members for attack and defense values
- Constructor that assigns unique values to type, armor, and strength points and seeds a random number generator
- Override attack function
- If attack roll is equal to 12, then Medusa uses Glare to increase attack value to 1000.

Blue Men class

- Inherits from Creature class
- Private data members for attack and defense values
- Constructor that assigns unique values to type, armor, and strength points and seeds a random number generator
- Override defense function
- For every 4 points of strength points lost, Blue Men uses Mob and loses 1 defense die

Harry Potter class

- Inherits from Creature class
- Private data members for attack and defense values
- Constructor that assigns unique values to type, armor, and strength points and seeds a random number generator
- Override damage function

- If strength points were equal to or less than 0, then Harry Potter uses Hogwarts and revives with its strength points equal to 20. This only happens once per game.

Arena function

- Takes in two Creature pointers through its parameter
- Player 1 always has the first turn
- Displays values for attack and defense function for each player turn
- Displays remaining strength points at the end of each round
- Fight continues until one creature has a strength point equal to or less than 0

Tournament class

- Arena function prototype
- 3 structures declared(Team A, Team B, and Loser Pile)
    - Contains 1 Creature pointer, 1 next pointer, and constructor
- First and last pointer for Team A and B and top pointer for Loser pile
- Integers for accumulated points and tracker for number of elements
- Constructor to set pointers to NULL and integers to 0
- Destructor deallocates the 3 structures
- Functions for adding, displaying, removing, swapping Creature pointers
- Function to call arena function, outputs the winner, add loser to the Loser Pile and remove from team, winner recovers strength

points and is places in back of node, accumulates points till one team has no Creature's left , and determine which team is the winner while displaying the Loser Pile

Menu function

- Create two Creature pointers
- Create Tournament object
- Displays menu list
- Displays different creatures to pick from
- Prompts user to pick which creature to add for each team
    - Asks if user wants to change the name of the selected creature
- Prompts user to display creatures in Team A and Team B
- Calls fight member function from Tournament object and displays losers

Main function

- Calls menu function

Assumptions made:

- More than 1 Creature object per team
- Both Teams will have same numbers of creature objects
- 1 test run for program

Problems that occurred:

There were two main problems that occurred for me during making this program. One of them was getting the outputted attack and defense values to match the outputted damage calculation. This problem was solved by putting the attack and defense function in an integer variable and using that variable through the damage function. Another problem was displaying my loser pile. I resolved this issue by placing a counter in my add loser pile function and using that counter for my displaying loser function.

Test Plan:

Test Case:	Input Values	Expected Outcome	Actual Outcome
Both Teams have 0 members	n/a	Segmentation Fault	Segmentation Fault
Both Teams have 1 member	1 Creature object vs 1 Creature object	Programs runs successfully and outputs winner	Programs runs and gets error when trying to swap nodes
Both Teams have 5 members	5 Creature object vs 5 Creature object	Programs runs successfully and outputs winner	Programs runs successfully
Team A has 2 members and Team B has 3 members	2 Creature object vs 2 Creature object	Programs runs and skews results to favor Team B	Programs runs and gets error if Team A wins