Kais Saradi
04/30/2019
Project # 2

Reflection Document

Design:
Animal Class- The animal class will contain all the set and get functions. It will be initialized with values of 0 for the required variable members of the class. This will be the parent class for the tiger, penguin, and turtle functions.
Tiger, penguin, and turtle classes - these classes will be the child classes for the animal class, so they will inherit all the member functions. The classes will have initializing constructors which will create a new tiger, penguin, or turtle object, using mutators for the private members of the animal class to initialize the animal.
Zoo class - The zoo class will contain all the functions needed to implement the game. It will initialize a zoo object, set the amount and capacity for each animal, create the dynamic array, and set the total account balance and profit. It should also have functions to add and remove animals, which will resize the dynamic array if necessary and add or remove the animal to it. There should also be a function to move to the next day, and generate a random event. For the random day event, there should be two other functions necessary. One will remove an animal from the zoo at random, and one will generate a baby at random. These function will do all the checking each exhibit for an animal to remove or have babies, and will be called if a 1 or 3 is rolled. Finally, there should be set and get methods for the balance and profit.
Menu - The menu will contain the input validation function from last weeks project, as well as a start menu which can be called to start the game.
Main - The main function should call the menu function, then start a game. When the game is started, the main function will ask the prompt the user for the initial conditions, initialize the zoo object and animal arrays, and starts at day 1 in a while loop, print the balance, change the age of the animals, then it will call the random event, calculate the profit, ask the user if they want to add an animal, and if they would like to continue playing. If the user chooses to stop playing the while loop ends, and the final message shown.

Reflection:

An issue I had with the last lab was getting valid input. After emailling the TA, I looked into using getline instead of cin.ignor and clear to validate input. Another issue I had was removing an animal when there were none. At first I was using a void function, which would show no animals in the exhibit no matter what I trie. I solved this by creating a bool function which would return false if there was already no animals in the exhibit, and used that in conjunction with the randomDeath() function to check each exhibit.

With the random function, I was successful in creating the randomDeath function to check the exhibit and remove and animal, but to add a baby I was not able to create a working seperate function, so included everything within the If statement in the randomEvent function. Finally, I struggled with memory leaks. It looks like as of turning in the project it still leaks memory. I will continue to debug, but memory leaks seem to be something I need to work on fixing.

Test Case	**Input Value**	**Driver Functions**	**Expected Outcome**	**Observed Outcome**
Integer error (invalid data type)	h	Menu error while loop	"Please enter an integer"	"Please enter please enter an integer"
Integer error (out of range)	0, 5	Menu while loop	"Please enter an integer (1 or 2)"	"Please enter an integer (1 or 2)"
Adding adult animal	1, 1 (tiger)	addtiger() function	"Animals in exhibit Tiger : 2"	"Animals in exhibit Tiger : 2"
Removing animal	N/A (1,1,1,1)	randomEvent(), randomDeath()	"A Turtle is removed from the zoo"	"A Turtle is removed from the zoo""
A baby is born	N/A (1,1,1)	addPenguinChicks()	"5 Penguin chicks were hatcked!... Animals in exhibit Penguin: 6"	"5 Penguin chicks were hatcked!... Animals in exhibit Penguin: 6"
Exit	1,1,1,2	N/A	"Final balance : x Goodbye!"	"Final balance : 9995590 Goodbye!"