Design

Main

Have minimal functions try to keep as clean as possible and pull from classes.

Animal Class

Since this is the parent class have all variables stored in here and have getter and setter functions for all of them.

Tiger Class

Have the default constructor set all the member variables per the rubric.

Bear Class

Have prompts before the main game starts where the user can enter information for this class. This should be the bonus class.

Penguin Class

Have the default constructor set all the member variables per the rubric.

Turtle Class

Have the default constructor set all the member variables per the rubric.

Zoo Class

I would like this to be the heart of the program with program ultimately taking it's directions from here.

The following function should be contained:

- Play/quit
    - Either starts or quits the game based on the users choice.
- Prompts for setting up the bear class (the added class)
    - Set price, number of kids, cost to feed, and profit percentage
- Starting menu
    - Loop through for each type of animal and ask what the user would like to buy.
- Set feeding type
    - Sets the type of feed and makes the correct adjustments to feed cost
- Random Event
    - Take into account feed type and adjust event probability accordingly
    - Trigger
        - Baby born
        - Boom in zoo attendance
        - Animal dies
        - Nothing event
- Baby born

    - Account for number of parents and situation if there aren't any parents
- Boom in attendance
    - Determine bonus and add to bank account
- Animal Dies
    - Select probability and remove animal
- Display profit
    - Tally profit from the day and display as well as remove from bank account
- Food cost
    - Show food cost and remove cost from bank account
- Prompt to buy a new animal
    - Ask which type
    - Add that animal
    - Deduct cost of animal
- End the day
    - Increase day variable as well as each animals age
    - Reset food type cost for next day.

such as when to play and exit. I'd also like to have all the functions such as adding and removing animals as well as random events etc etc.

TESTING

Test	Description	Expected Result
Input validation for menu, and choice throughout the game.	Test to see if player typed acceptable input.	If inappropriate entry is made message will display indicating input was invalid and prompt to enter again.
Play or quit	Decision tree based on if the user wants to play or quit	If play take to main menu if quit, quit program
Random event	Random events are within range	Correct even chosen for number generated based on feed type.
Possible birth	Check to see if there is at least 1 adult	If there is one animal of that type increase the number of adults by one.
No adults for that class	Move to next type of animal	Move to the next animal down the list
No adults at all	Display alt reward	Give owner a bonus for cute animals.
Food cost	Calculate food cost based on food type	Calculation done correctly for both food type and type of animal.
Purchase animal	Animal is purchased and price deducted	Increase number of animals you own by one and deduct the cost for that animal from the bank.

Reflection

This project has really shown me that my planning skills as to how to design these programs needs much improvement. At the start of the planning phase I went way to deep down the rabbit hole into the very tiniest of minute details and over engineered my program ultimately not accounting for road blocks that I would face later. Ultimately this resulted in me starting from scratch mid way through the project with a better more adaptable plan. I do however appear to continue to be having trouble with memory leaks. I've spent the past 8 hours now looking at the report and for the life of me I cannot figure out what is wrong. I'm going to have to start doing this as I build the program and just call destructor functions throughout the way in an attempt to try to narrow down where I've gone wrong.

I also feel that while I'm an okay programmer in that given enough time I'll eventually come up with a solution it became increasingly apparent how elementry my solutions are when doing the code review. I noticed some of the other students didn't use nearly the number of if then statements that I have found in my program. There were a couple of times throughout this project where after it was built I changed it such as having an if then statement when asking how many animals to buy.

If (1)
Add tiger
Else if (2)
Add tiger
Add tiger

While this works, I can see how it's extremely cumbersome on the system rather than having a for loop with a counter which is what I ended up using. I'd ultimately like to improve on this in using a lot less if then statements in my code as I feel it's become overly complex and there are much simpler ways to accomplish what I am trying to do. I would appreciate your help if you could point out anything that you see is glaringly obvious in this program and offer a suggestion. I'd also like to ask if you could tell me where I went wrong with the memory leak I have been searching now for approx 7 hours reworking destructors, moving functions that destroy arrays etc etc and nothing I do seems to help.