Matthew Solbrack
CS 162/400
Project 2

Main Design:
The Instructions gave us the classes that will be used in this project. As well as instructions about some member variables and the use of inheritance. Here is what we know so far:

- class: Animal
  member Variables: age, cost, numberOfBabies, baseFoodCost, payoff
  Virtual Functions: setCost, setNumberOfBabies, setPayOff, setBaseFoodCost
  getFunctions: getAge, getCost, getNumberOfBabies, getPayOff, getBaseFoodCost
- class: Penguin
  Override functions: setCost, setNumberOfBabies, setPayOff, setBaseFoodCost
- class: Tiger
  Override functions: setCost, setNumberOfBabies, setPayOff, setBaseFoodCost
- class: Turtle
  Override functions: setCost, setNumberOfBabies, setPayOff, setBaseFoodCost
- class: Zoo
  Member Variables: tiger, penguin, turtle (dynamic arrays). int for keeping track of the bank, the day, and the profit, we will need an animal pointer for use of the inheritance to the specific animals.

I will use my own classes to simplify some aspects of the Menu and Input Validation

- menu
- checkInput

The Zoo class:

- Major functions that will be needed include, gameplay, add one year to each animal, random event function, increase the size of the array function, purchase additional animals function, feeding cost function, zoo admission function, constructor to initialize the variables and destructor to delete the dynamic arrays.

Test Table

Test Case	Input Values	Expected Outcomes	Observed Outcomes
Input Negative	-1, -22, -5	"Try Again" or Brings you back to the original menu or question	"Try Again"
Input at 0	0	"Try Again"	"Try Again"
Input in Current Range	When Prompted: 1 or 2 When Prompted: y or n When Able: 3,4,5,6	Should go through the program and bring you to the menu to run the simulation again and again and again	The game either kept going and eventually ran out of money and the game ends.
Input low	2, 3, 4, 5	Should go through the program and bring you to the menu to run the simulation again.	Some of the data didn't get excepted the cause it was out of scope "Try Again" some of the data kept the game play going and going and going.
input extremely High	1000, 99, 55	The only time this will be excepted is when prompted to buy additional animals. You should run out of money.	The only time this data was excepted is when prompted to buy additional animals. all other inputs were "Try again". When entered into the additional animals prompt you run out of money and the game ends.
other than integer	#, W,	"Try Again"	"Try Again"

Reflection

I feel like I could just keep going on this project forever to bug out everything. It's hard to understand the dynamics of dynamic arrays. I feel like they are very unstable. Also, I could not figure out how to resize the same array multiple times. I looked online for hours and I could not find the information to solve the issue.

Some changes I made to the original plans came mostly in the destructor. I had a difficult time figuring out how to delete the dynamic arrays after they were resized. I added resizes dynamic variable for each animal as a member variable, I also added bool variables to see if the resize variable was used. This seemed to work out the bug that was not allowing me to end the program.

Another aspect I changed was keeping track of the number of animals in the zoo. I don't know if this was a good idea or not. But, it made it easier instead of calling the dynamic pointer array every time.

Overall, I am happy with the result of this project. This is the largest program I have built to date. It was challenging and fun at the same time.