**Project Design**

The Zoo Tycoon game should have six required classes; the zoo class, the animal class, the tiger class, the penguin class, the turtle class, and the NewAnimal class. The Animal class will be a pure virtual function that will be the base class for the tiger, penguin, turtle, and NewAnimal classes. Within the zoo class, there are five required member variables. Of the five, four of them will be overridden by the derived class, as they have different values for the each of those classes.

The Zoo class will house the main variables and functions for the project. It will have a variable to hold the amount of money the zoo currently has, dynamic arrays for each of the animal types, and a function that will run the game. The game should ask the user if they want to continue to the next day at the end of every loop. In that loop, the age of each animal will be increased, they will be fed, which means that we reduce our cash amount, and we determine how much we made from the payoff, which increases our cash.

A random event then occurs, which is determined by a separate function. Each of these random events has an equal chance to occur. The three random events are one of the species getting sick, one of the species having a baby, and an attendance boom.

The game repeats until the user ends the game.

**Pseudocode**

Animal class

Private:

Int age;

Double cost;

Int numBabies;

Double baseFoodCost;

Double payoff;

public:

Animal();

~Animal();

// Getters

Int getAge();

Double getCost();

Int getNumBabies();

Double getBaseFoodCost();

Double getPayoff();

// Setters

Void setAge(int ageIn);

Void setCost(double costIn);

Void setNumBabies(int babiesIn);

Void setBaseFoodCost(double foodCostIn);

Void setPayoff(double payoffIn);

// increase age

Void increaseAge();

These are repeated for the four derived classes.

Zoo class

Private:

Double cash;

Int currentDay;

// Tiger array

Tiger \* tigerArray;

// Penguin array

Penguin \* penguin Array;

// Turtles array

Turtle \* turtleArray;

// NewAnimal array

NewAnimal \* newAnimalArray;

Public:

Tiger\* getTigers; // returns the tigerArray

Penguin\* getPenguins; // returns the penguinArray

Turtle\* getTurtles; // returns the turtleArrya

NewAnimal\* getNewAnimals; // returns the newAnimalArray

runGame(); // function to run the game

ageAnimals(); // to age the animals at the beginning of the day

feedAnimals(); // To feed the animals and decrement cash

performRandomEvent(); // Does a random event

welcomeMessage(); // Intro message

endDay(); // Message sent out at the end of the day

addTiger(tigerIn); // Adds a tiger to the array

addPenguin(penguinIn); // Adds a penguin to the array

addTurtle(turtleIn);// Adds a turtle to the array

addNewAnimal(newAnimalIn); // Adds a newAnimal to the array

sickAnimal(); // Kills one of the animals

attendanceBoom(); // Increases the cash

babyBorn(); // Adds a new animal to a random species

calculateProfit(); // Calculates how much the zoo made

checkBalance(); // Checks the banks balance

**Test Tables**