Carrie Davis

Intro Programming 162

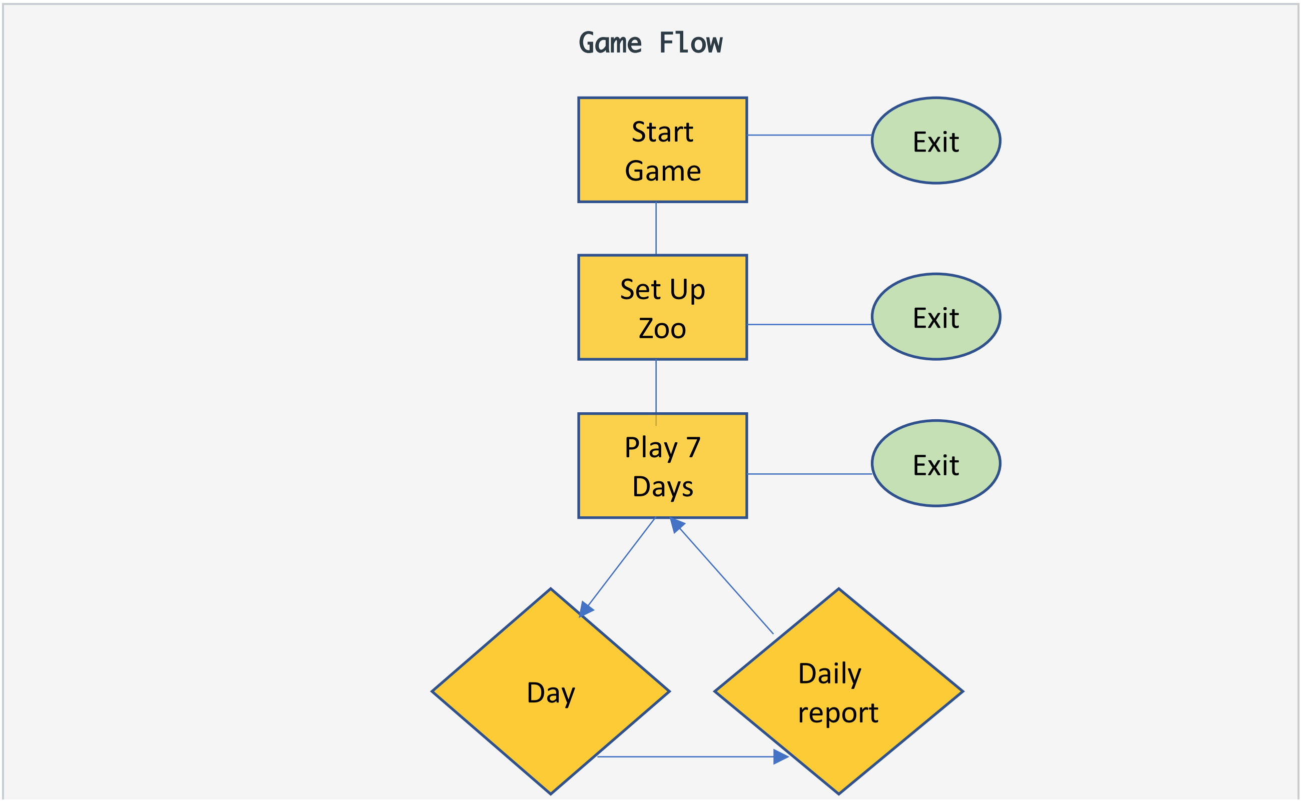
Project 2 Plan

Program Requirements/Steps

Zoo Tycoon with Animal Inheritance Design

Create a program that carries out following steps.

1. Requests from user if they want to start program or exit.
2. Starts Game with choice 1.
3. Sets up Zoo Array and bank total to $100000.
4. Prompts user to purchase 1 or 2 of each type of animal.
5. Charges for animal and sets animals into array at 1 day old.
6. Asks user if they would like to start week 1 or exit.
7. Loops through 7 days. Each day animals add to age +1, pay food cost, profit, random act happens.
8. Displays report for each day and each anima in zoo.
9. Displays bank balance for that week.
10. Prompts user to run another week or exit.

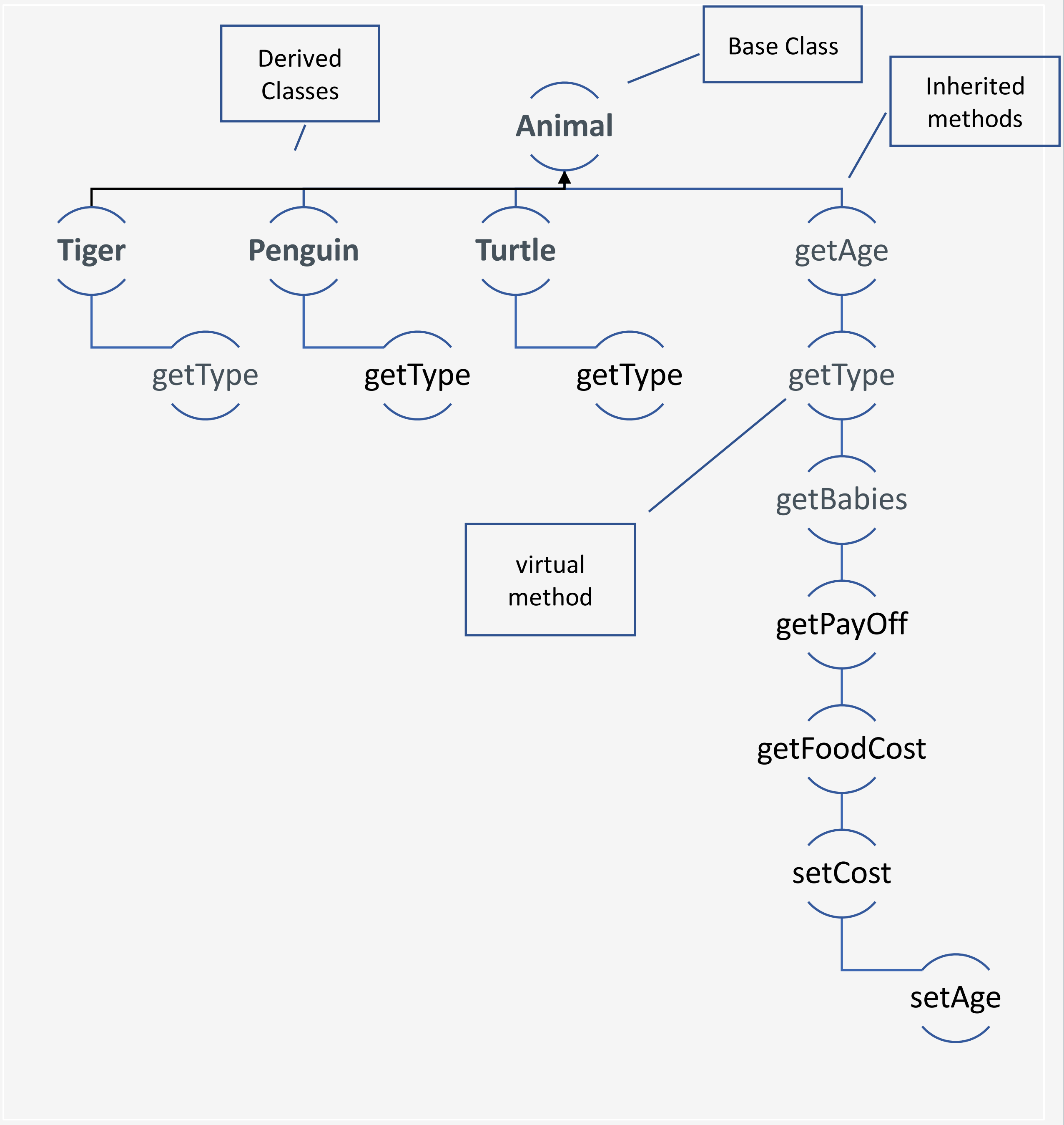


```
Start Game: main()
Prompt user if start program or quit
input user choice
1 → start program
2 → Do nothing and quit
#%* or != 1 || 2  → while choice is not 1 or 2 prompt user to enter 1 or 2
ZooTycoon() play method
- startZoo
- Prompts user to start 1st week
        input user startchoice
1 → start first week
2 → Do nothing and quit
#%* or != 1 || 2  → while choice is not 1 or 2 prompt user to enter 1 or 2
Starts week loop
int choice // choose whether to do another week
int week = 1 // start with week 1

do{ for (each day < 7, add day)
            {- day();
            - displays day report}
      Displays weekly bankTotal;
        1 → continue another week
        2 → exit game
#%* or != 1 || 2  → while choice is not 1 or 2 prompt user to enter 1 or 2
            Week++;  // do another week } While (choice ==  1)

//when choice is not 1 free memory
      Delete [] zoo array;
```

```
ZooTycoon::startZoo()

        Prompt user to purchase 1 or 2 of tiger, turtle and penguin.

        Input user # of each animal type.

        int tigerAmount, penguinAmount, turtleAmount;

        1 -> one animal

        2 -> two animals

#%* or != 1 || 2  → while choice is not 1 or 2 prompt user to enter 1 or 2

        Displays cost and bankTotal update.

        Creates dynamic zoo array with different animal types.

        Adds to the numberOfAnimals data member to keep track of animal amount
.

ZooTycoon::day()

        for (each animal in zoo array)

                -Display Animal Type

                -Display daily cost

                -Display profit made from animal

        Random

                4 cases that are randomly chooses using random_device

        Switch (random choice)

                Case 1: sinkness(); // one animal will die, removed

                        break;

                Case 2: for (each tiger in zoo array)

                        bonus(); //bonus is random between 250 – 500

                        break;

                Case 3: babyBorn(); //random baby based on animal age

                //Number of babies based off baby data member for animal type

                Case 4: nothing happens. Output message.

                        Break;
```

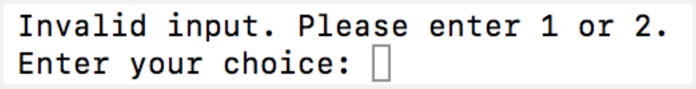
Animal Specification

Inheritance

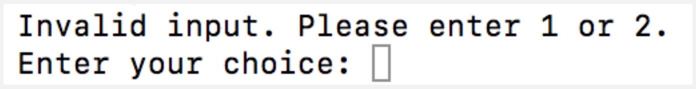


Test Plan

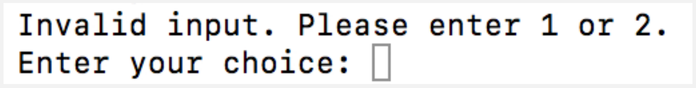
Tests	***Action performed***	***Expected output***

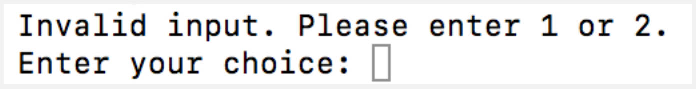
Test1:	tigerAmount
	Entered 0
		
		`Invalid input. Please enter 1 or 2.`
		`Enter your choice:`

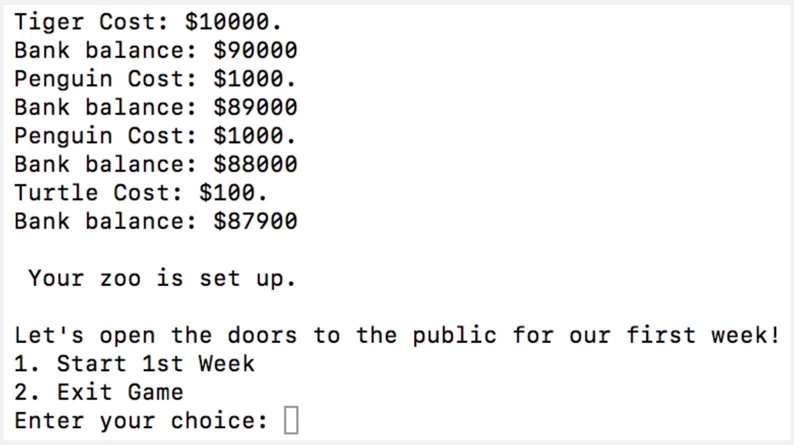
	tigerAmount Entered 1	**Animal:** 1 “TIGER”

	penguinAmount
	Entered 100
		
		`Invalid input. Please enter 1 or 2.`
		`Enter your choice:`

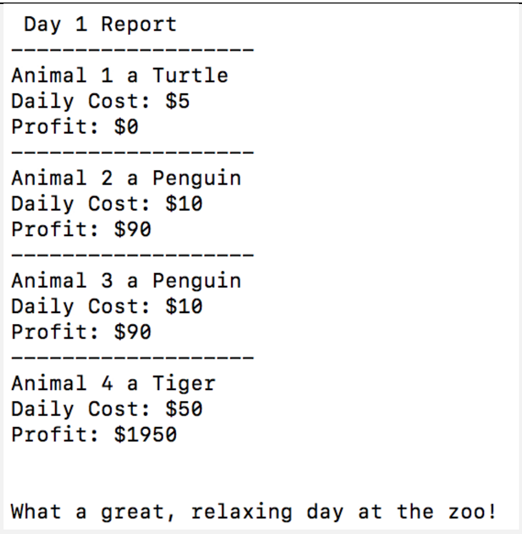
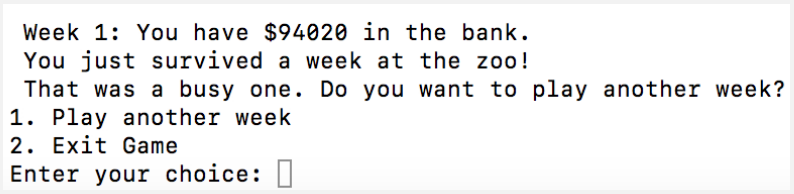
	penguinAmount Entered 2.2	**Animal:** 2 “PENGUIN”

	TurtleAmount
	Entered %$4
		
		`Invalid input. Please enter 1 or 2.`
		`Enter your choice:`

	TurtleAmount
	Entered %$2
		
		`Invalid input. Please enter 1 or 2.`
		`Enter your choice:`

	TurtleAmount
	Entered 1.1
		Animal: 1 “TURTLE”
		Print Out Result Example
		
		`Tiger Cost: $10000.`
		`Bank balance: $90000`
		`Penguin Cost: $1000.`
		`Bank balance: $89000`
		`Penguin Cost: $1000.`
		`Bank balance: $88000`
		`Turtle Cost: $100.`
		`Bank balance: $87900`
		
		` Your zoo is set up.`
		
		`Let's open the doors to the public for our first week!`
		`1. Start 1st Week`
		`2. Exit Game`
		`Enter your choice:`

	startChoice Entered 1.4$	**startChoice:** 1 **Print Out Result Example for 1st Day** **note random sickness, baby born, tiger bonus will not always be same.*

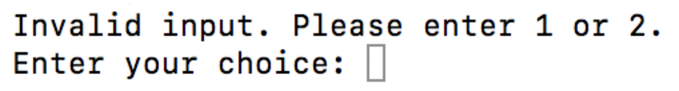
	choice Entered
	2.2
		
		` Day 1 Report`
		`--------------------`
		`Animal 1 a Turtle`
		`Daily Cost: $5`
		`Profit: $0`
		`--------------------`
		`Animal 2 a Penguin`
		`Daily Cost: $10`
		`Profit: $90`
		`--------------------`
		`Animal 3 a Penguin`
		`Daily Cost: $10`
		`Profit: $90`
		`--------------------`
		`Animal 4 a Tiger`
		`Daily Cost: $50`
		`Profit: $1950`
		
		`What a great, relaxing day at the zoo!`
		
		… 6 more Days * note bank total is due to random bonuses and may change due to random cases
		
		
		` Week 1: You have $94020 in the bank.`
		` You just survived a week at the zoo!`
		` That was a busy one. Do you want to play another week?`
		`1. Play another week`
		`2. Exit Game`
		`Enter your choice:`
		
		choice: 2
		
		Game will end

Test 2:

tigerAmount
Entered 2.24645

Animal: 2 “TIGER”

penguinAmount
Entered 500.5



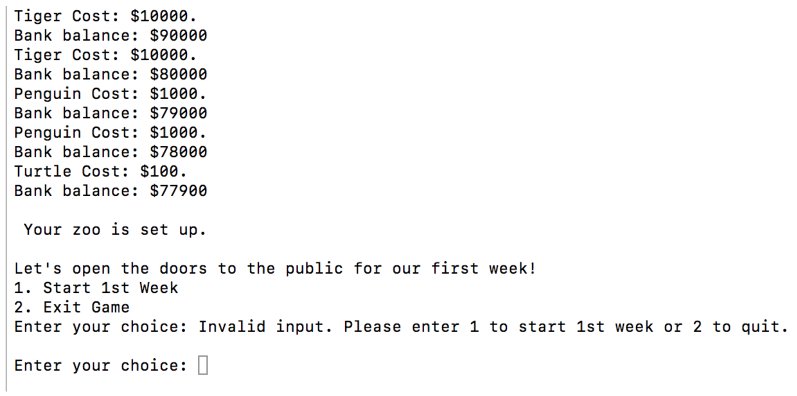
```
Invalid input. Please enter 1 or 2.
Enter your choice:
```

penguinAmount
Entered 2.8

Animal: 2 “PENGUIN”

Animal: 1 “TURTLE”

TurtleAmount
Entered 1@

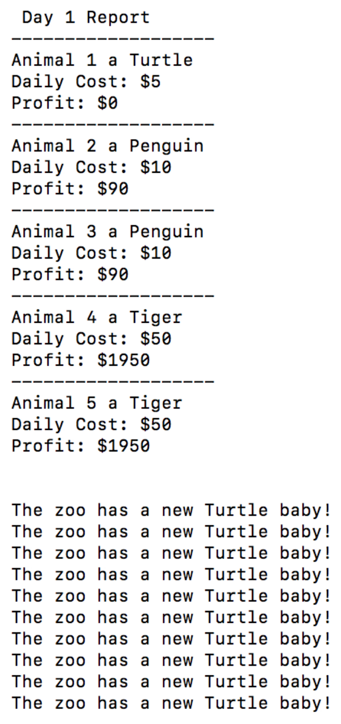


```
Tiger Cost: $10000.
Bank balance: $90000
Tiger Cost: $10000.
Bank balance: $80000
Penguin Cost: $1000.
Bank balance: $79000
Penguin Cost: $1000.
Bank balance: $78000
Turtle Cost: $100.
Bank balance: $77900

 Your zoo is set up.

Let's open the doors to the public for our first week!
1. Start 1st Week
2. Exit Game
Enter your choice: Invalid input. Please enter 1 to start 1st week or 2 to quit.

Enter your choice:
```

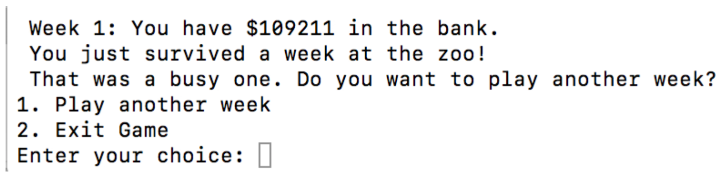
startChoice
Entered 1.4$

startChoice: 1

Print Out Result Example for 1st Day **note random sickness, baby born, tiger bonus will not always be same.*



```
 Day 1 Report
-------------------
Animal 1 a Turtle
Daily Cost: $5
Profit: $0
-------------------
Animal 2 a Penguin
Daily Cost: $10
Profit: $90
-------------------
Animal 3 a Penguin
Daily Cost: $10
Profit: $90
-------------------
Animal 4 a Tiger
Daily Cost: $50
Profit: $1950
-------------------
Animal 5 a Tiger
Daily Cost: $50
Profit: $1950


The zoo has a new Turtle baby!
The zoo has a new Turtle baby!
The zoo has a new Turtle baby!
The zoo has a new Turtle baby!
The zoo has a new Turtle baby!
The zoo has a new Turtle baby!
The zoo has a new Turtle baby!
The zoo has a new Turtle baby!
The zoo has a new Turtle baby!
The zoo has a new Turtle baby!
```

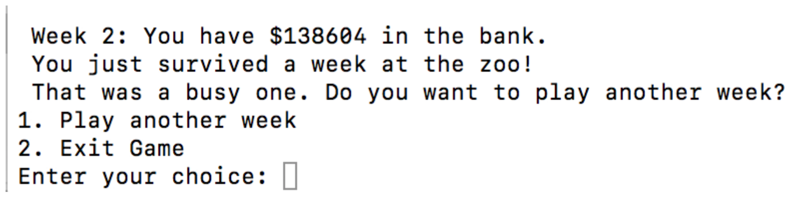


```
 Week 1: You have $109211 in the bank.
 You just survived a week at the zoo!
 That was a busy one. Do you want to play another week?
1. Play another week
2. Exit Game
Enter your choice: 
```

startChoice Entered 1.4#

startChoice: 1

Print Out at end of week*note random sickness, baby born, tiger bonus for each of 7 days bankTotal will not always be same.*

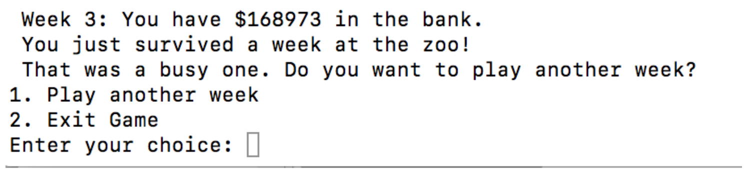


```
 Week 2: You have $138604 in the bank.
 You just survived a week at the zoo!
 That was a busy one. Do you want to play another week?
1. Play another week
2. Exit Game
Enter your choice: 
```

startChoice: 1

startChoice Entered 1^4

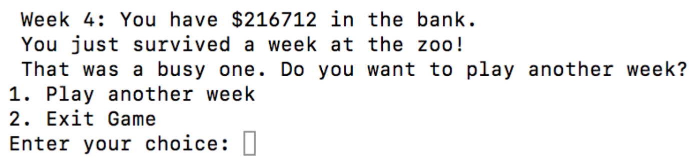
Print Out at end of week*note random sickness, baby born, tiger bonus for each of 7 days bankTotal will not always be same.*



```
 Week 3: You have $168973 in the bank.
 You just survived a week at the zoo!
 That was a busy one. Do you want to play another week?
1. Play another week
2. Exit Game
Enter your choice: 
```

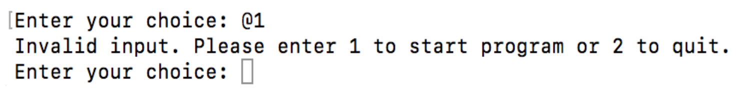
startChoice Entered 1.8

startChoice: 1



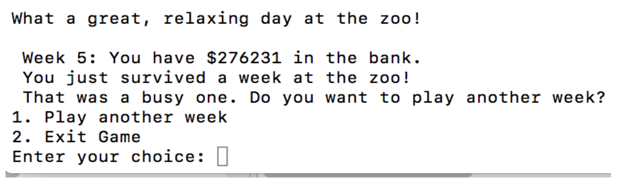
```
 Week 4: You have $216712 in the bank.
 You just survived a week at the zoo!
 That was a busy one. Do you want to play another week?
1. Play another week
2. Exit Game
Enter your choice: 
```

startChoice Entered @1

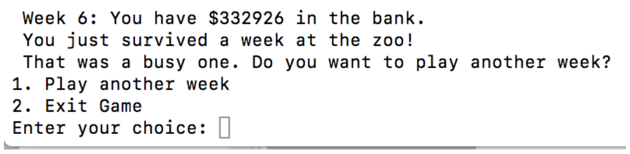
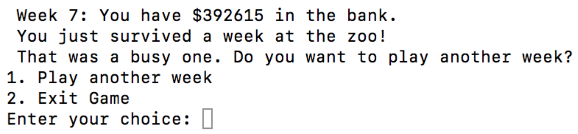
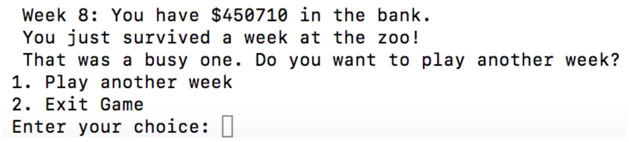
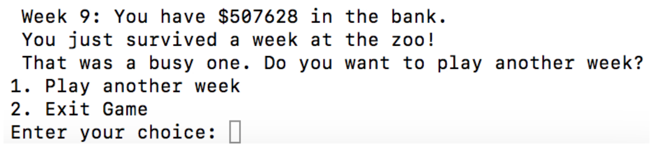
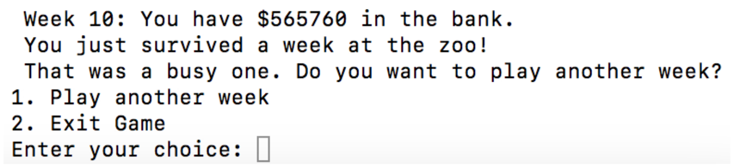
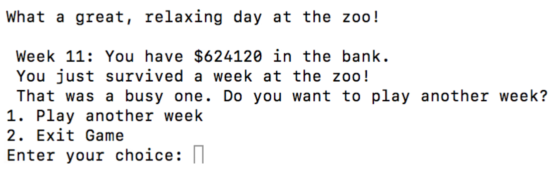
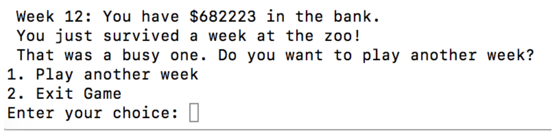
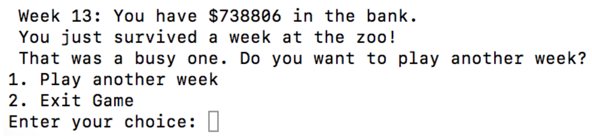


```
Enter your choice: @1
Invalid input. Please enter 1 to start program or 2 to quit.
Enter your choice: 
```

startChoice Entered 1



```
What a great, relaxing day at the zoo!

 Week 5: You have $276231 in the bank.
 You just survived a week at the zoo!
 That was a busy one. Do you want to play another week?
1. Play another week
2. Exit Game
Enter your choice: 
```

	startChoice Entered 1	 Week 6: You have $332926 in the bank. You just survived a week at the zoo! That was a busy one. Do you want to play another week? 1. Play another week 2. Exit Game Enter your choice:
	startChoice Entered 1	 Week 7: You have $392615 in the bank. You just survived a week at the zoo! That was a busy one. Do you want to play another week? 1. Play another week 2. Exit Game Enter your choice:
	startChoice Entered 1	 Week 8: You have $450710 in the bank. You just survived a week at the zoo! That was a busy one. Do you want to play another week? 1. Play another week 2. Exit Game Enter your choice:
	startChoice Entered 1	 Week 9: You have $507628 in the bank. You just survived a week at the zoo! That was a busy one. Do you want to play another week? 1. Play another week 2. Exit Game Enter your choice:
	startChoice Entered 1	 Week 10: You have $565760 in the bank. You just survived a week at the zoo! That was a busy one. Do you want to play another week? 1. Play another week 2. Exit Game Enter your choice:
	startChoice Entered 1	 What a great, relaxing day at the zoo! Week 11: You have $624120 in the bank. You just survived a week at the zoo! That was a busy one. Do you want to play another week? 1. Play another week 2. Exit Game Enter your choice:
	startChoice Entered 1	 Week 12: You have $682223 in the bank. You just survived a week at the zoo! That was a busy one. Do you want to play another week? 1. Play another week 2. Exit Game Enter your choice:
	startChoice Entered 1	 Week 13: You have $738806 in the bank. You just survived a week at the zoo! That was a busy one. Do you want to play another week? 1. Play another week 2. Exit Game Enter your choice:

Reflection

Setting up this project I started with Animal.cpp, Animal.hpp and the man.cpp. Testing all the methods with Animal instances first. Then test adding classes for tiger, penguin and turtles. I placed all the class information for each of these in the Animal.hpp. It seems like for organizational purposes a waste to create separate cpp/hpp files right now when the derived classes only have one class that is overridden. Example of how tiger class is defined within the Animal.hpp. The Turtle and Penguin classes are similarly structured.

```
class Tiger: public Animal{

public:

    //derived all functions from animal

    Tiger(int age,int baby,int foodCost,int payOff): Animal(age,baby,foodCost
,payOff)

    {

    }

    TYPE getType(){ return TIGER;}

};
```

Then I based my zoo on a Dynamic array that pointed to Animal pointers. This is defined in the zooTycoon::startZoo() method. This is similar to how I created my Die pointer array in Lab 3.

```
ZooTycoon.hpp //in private data members

    int numberOfAnimals; // size of animal array

    Animal** animalArray; //animal array


ZooTycoon.cpp line 179

  numberOfAnimals = tigerAmount + penguinAmount + turtleAmount;

    animalArray= new Animal*[numberOfAnimals];
```

I did need to recreate this array when a sickness removed an animal, or a birth added one. I used information from Chapter 8 Copying Array to and the article https://www.cs.fsu.edu/~myers/c++/notes/dma.html information to think of how to copy, delete and recreate my animalArray after a death or birth of an animal.

One error I encountered early which seems to always happen when I try using dynamic pointers is that I received a “segmentation fault” error. This time because a misplaced counter made my array go out of the SIZE bounds when looped.

```
    Original -181 ZooTycoon.cpp
      animalArray= new Animal*[numberOfAnimals];
    int counter = numberOfAnimals-1;



    for (int tiger = 0; tiger < tigerAmount; tiger++)
    {
        counter--;
        bankTotal -= 10000;
        animalArray[counter] = createAnimal(TIGER);
    }
…..  Fixed


   for (int tiger = 0; tiger < tigerAmount; tiger++)
    {


        bankTotal -= 10000;
        animalArray[counter] = createAnimal(TIGER);
        counter--;
```

I also tested different methods for randomizing selections. Tried creating a standard method to use but it did not fit all of my scenarios. For example I used the following for picking any animal for sickness and for the random day events.

```
ZooTycoon.cpp 305
       int randAnimal;
    //randomly pick animal from array
    random_device rd; //seed
    mt19937 gen(rd()); // standard mersenne_twister engine
    //seed value initalization and randomization
    int maxSeed = numberOfAnimals-1;
    int minSeed = 0;

    uniform_int_distribution<int> dist(minSeed,maxSeed);
    randAnimal = dist(gen);
```

Then I used random_shuffle for determining which animal >= age 3 would have babies. Because I could use a range that would change based on the animals that are 3 or over and then randomly shuffle the array of index values of those animals to choose one `if the parentSubscript > 0`.

```
ZooTycoon.cpp 382
        int * parentList = new int[numberOfAnimals];
    int parentSubscript = 0;
    for(int animal = 0 ; animal < numberOfAnimals; animal ++ )
    {
        int age = animalArray[animal]->getAge();
        if(age >= 3) //this is the age where an animal can be a parent
        {
            parentList[parentSubscript] = animal; //represents the ind
ex of a possible parent animal

            parentSubscript++;

        }

    }

if(parentSubscript > 0)

    {

    random_shuffle(&parentList[0], &parentList[parentSubscript]); // s
huffle values then take first



    int randomIndexValue = parentList[0]; // random index value that r
epresents a possible parent animal
```

I did encounter some questions on how to input my pointer int array and used the following article for advice.
 https://stackoverflow.com/questions/14720134/is-it-possible-to-random-shuffle-an-array-of-int-elements

If I had more time I could clean this up and only recreate array for all the babies instead of each time a baby is born.

```
        ZooTycoon.cpp 402

          for(int baby=0; baby < numberOfBabies; baby++)

        {

            Animal* newAnimal = createAnimal(parentType);

            newAnimal->setAge(0);

            cout << "The zoo has a new "

<< getTypeString(animalArray[randomIndexValue]) << " baby!"<<endl;



            // add baby and recreate array

            Animal** temp = new Animal*[numberOfAnimals + 1];
            for(int animal = 0; animal < numberOfAnimals; animal ++)
            {   temp[animal] = animalArray[animal];}

            temp[numberOfAnimals] = newAnimal;

            numberOfAnimals = numberOfAnimals + 1;  // reset size to a
dd animal

            delete [] animalArray; //delete old array
            animalArray = temp;

        }
```

It was difficult for me to test the random_shuffle to make sure it would access the Tiger and Pengiun classes because my zoo easily becomes overrun with turtles. I used the get method and cout messages to make sure my shuffle was working and randomizing properly because it did seem that I always had more turtles. Another improvement would be to update the screen to accommodate and lump reports for turtles.

```
Testing in ZooTycoon.cpp 394 for randomness
for( int i = 0; i < parentSubscript; i++)
    { cout << parentList[i] << "this is the value for parent list afte
r shuffle " <<endl; }
```

The tests in this plan were also very helpful in validating my input validation code. Validation was an issue with iostream. I had the same problems as with Project 1 explained in this article: https://stackoverflow.com/questions/47957584/why-does-ifcin-int-accept-a-decimal-number-in-the-first-iteration-but-no. I did set some of my user inputs to double and did a static_cast to integer.

```
    double choice;
   cin >> choice;
    choice = static_cast<int>(choice);
```

Test 2 failed with 1@ even when using the value above for input. I was able to correct this by using cin.ignore() before the next cin input to “ignore” the @. To keep my next cin input from using this value and failing. For example.

```
  Main.cpp 41
   if(choice == 1)
    {
       //clear cin
        cin.ignore(std::numeric_limits<std::streamsize>::max(),'\n');
        //create game instance and start play
        ZooTycoon newGame;
        newGame.play();
```