Final Project Specification

* Models ecosystems that can have different species, which each have characteristics such as type(name), genetic traits, and population size(which changes over time)
* The project will have the four classes Ecosystem, Species, IndependentSpecies, and PredatorPrey. The IndependentSpecies and PredatorPrey classes and both derived from the Species class
* The user will be able to choose the number of species to analyze and store the information for the species in a vector. This is the main function of the Ecosystem class. The class has functions for adding species and for returning the number of species, independent species, and species in predator/prey relationships
* Species objects with data fields of name, size, pAlleleFrequency, and qAlleleFrequency will be made to represent species. There will be a no-arg constructor and a constructor in which the values for all the data fields are specified by the user. Information for the data fields can be read in from a file. All data fields are private and accessor functions are provided
* The program has functions that can calculate the percent of individuals with the pp (homozygous dominant) genotype, the percent with the pq (heterozygous) genotype, and the percent with the qq (homozygous recessive) genotype based on allele frequencies. It can also calculate both the number and percentage of individuals with the dominant (pp or pq) phenotype and the number and percentage of individuals with the recessive (qq) phenotype.
* The main assumptions are that the user knows or can estimate the allele frequencies and that the population is in equilibrium. This means that no mutations or natural selection is taking place, alleles are not being transferred to and from other populations, mating is random, and the population size is large. The program will have a function that can determine whether a population is in equilibrium if genotype frequencies are known
* The additional data fields carryingCapacity(K), time(t), and growthRate(r), are included in the IndependentSpecies class. Three constructors are provided. One is a no-arg constructor, one takes in specified values for all the data fields in the Species and IndependentSpecies classes, and one takes in values for all values except for pAlleleFrequency and qAlleleFrequency, to be used if only population sizes and not genetic characteristics are being analyzed. Specified values for data fields can be read in from a file. The data fields are private and accessor functions are provided
* The class will have a function to calculate population sizes at a future time using the equation y = , which is the solution to the logistic equation. The program will also be able to store population sizes at different times in a vector and display the values for time and population size in a table
* One assumption is that the user knows the values for initial size, growth rate, and carrying capacity. The equation used also assumes that the only factors affecting population size are the initial size, the growth rate, and the carrying capacity.
* The program will have a PredatorPrey class with private data fields predatorName, predatorSize (, preyGrowthRate (a), predatorDeathRate (c), time (t), alpha (α), and gamma (γ). Accessor functions will be available for all data fields. Three constructors will be provided: a no-arg constructor, a constructor that takes specified values for name, time, growthRate, size, pAlleleFrequency, and qAlleleFrequency, and a constructor that takes specified values for name, predatorName, predatorSize, time, growthRate, deathRate, alpha, and gamma. Since there is not a constructor that takes specified values for all data fields in the Species and PredatorPrey classes, mutator functions will be provided for predatorName, predatorSize, predatorDeathRate, alpha, and gamma
* Functions will be provided to calculate the population sizes of both prey and predator at specified times using approximated solutions to the Lotka-Volterra equations: x = ) and y = . K and depend on initial conditions and the program will be able to solve for them.
* The assumptions are that the only factors affecting population sizes are those given in the data fields (initial population sizes, prey growth rate, predator death rate, and the amount of interaction between the species) and that each prey species has only one predator and each predator has one kind of prey. The program also assumes that the user knows initial population sizes, birth and death rates, and alpha and gamma. Also, for the approximated equations to model population sizes well, the prey population size should be close to (predator death rate / gamma) and the predator population size should be close to prey growth rate / alpha)
* The program will include at least one test case with preset values and at least one in which the user inputs a file