CSC1322 Advanced Programming in C++

Programming Assignment: Monitoring Rhinos: Part I

Objectives

- Be familiar with object-oriented programming concepts.
- Be able to implement and test classes

Introduction

This assignment is intended to familiarize you with C++ classes and objects. You will learn how they can be used to maintain data like a tracking system. You are about to write programs to keep track of an endangered species, rhinos. For more information about rhino, you can go to the website `www.wikipedia.com/rhinoceros`

What To Do

Create a Rhino class and declare the following class member variables (you can choose the names), which are meant to hold information describing a rhino. Make all these variables private and properly comment them. Here is an example of a declaration with a suitable comment:

```
char gender; // the gender of this rhino. ‘M’ for male, ‘F’ for female
```

1. nickname (a string): name given to this rhino, the length of the string name >0.
2. year of birth (an int): year the rhino was born. It must be >1900.
3. month of birth (an int): month the rhino was born in range 1..12, with 1 meaning January.
4. gender of this rhino (a character): ‘M’ means male and ‘F’ means female.
5. this rhino’s tag (an int): a positive integer. -1 means that this rhino has no tag yet.
6. mother (a rhino): the mother of this object. It should be declared as a pointer to a Rhino object. NULL if unknown (default value)
7. father (a rhino): the father of this object. It should be declared as a pointer to a Rhino object. NULL if unknown (default value)
8. number of children of this rhino

There are three groups of functions you need to declare and implement. Work with one group at a time. Do not go on to the next group of functions until the group you are working on is thoroughly tested and correct.

The names of your functions must match those listed below exactly, including capitalization. The number of parameters and their order must also match: any mismatch will cause our testing programs to fail during grading, meaning that you will lose points. Parameter names will not be tested – change the parameter names if you want.

In this assignment, you may not use **if-statements** anywhere. They are not necessary. Submissions containing if-statements will cause point deductions.

Group A: The first constructor and all the getter functions of class Rhino.

Functions	Return Type
Constructor Description ✍ Rhino(string n, int y, int m, char g) Constructor: a new rhino with nickname n, birth year y, birth month m, and gender g. Its parents are unknown, and it has no children. Precondition: n's length is >0 and y >1900.	
Getter Function Description	
✍ getName() returns this rhino's nickname	string
✍ getYear() returns year this rhino was born	int
✍ getMonth() returns month this rhino was born	int
✍ getGender() returns this rhino's gender ('M' or 'F')	char
✍ isMale() returns verification result of "this rhino is male"	bool
✍ getTag() returns this rhino's tag (>0; -1 if none)	int
✍ getMother() returns a pointer to this rhino's mother (NULL if unknown)	pointer
✍ getFather() returns a pointer to this rhino's father (NULL if unknown)	pointer
✍ getNumChildren() returns the number of children of this rhino.	int

To test your class, you need to create a program with `main ()` function, sometimes called client program. Within the main function, consider testing the constructor as follows: create one rhino object using the constructor and then check, using the getter functions, that all member variables have the correct values.

Group B: The setter functions. When testing the setter functions, you will have to create one or more rhino objects, call the setter functions, and then use the getter functions to test whether the setter functions set the member variables correctly. Good thing is that you already tested the getters! Note that two of the setter functions may change more than one member variable; your testing program should check that all member variables that may be changed are changed correctly. For example, Suppose `p` is a Rhino object. when calling to `p.addMother(Rhino * mom),` `mom` becomes `p`'s mother and also `mom` has an additional child.

Setter Function Description
✍ setTag(int n) sets this rhino's tag to n. Precondition: n >0. ✍ addMother(Rhino * mom) adds mom as this rhino's mother. Precondition: mom is not NULL, mom is female, and this rhino does not have a mother. ✍ addFather(Rhino * dad) adds dad as this rhino's father. Precondition: dad is not NULL, dad is male, and this rhino does not have a father.

Group C: The second constructor.

A class could have more than one constructor to accommodate the situations where not all the information about a rhino is available at the time an object is being created. The following constructor will be used if all the information about a rhino is available.

✍ Rhino(string n, int y, int m, char g, int t, Rhino * mom, Rhino * dad)
Constructor: a new rhino with nickname n, birth year y, birth month m, gender g, tag t, mother mom, and father dad.
Precondition: n's length is >0, y >1900, m is the month number, in 1..12, g is 'M' for male or 'F' for female, tag t >0 (or -1 if not tagged), and mom and dad may not be NULL.

What To Submit

- Organize your program into three files: a header file with class declaration named rhino.h, a C++ file named rhino.cpp with class definition (implementation of all functions including constructors), and a file with main function (you choose a name for it, for example, myRhino.cpp) in which it contains your test programs.
- Test your class thoroughly. Make sure all the functions including constructors are tested.
- Zip the three files into a single zip file and upload to the Blackboard by the deadline with a naming format like fullName.zip, e.g. johnDoe.zip. Your programs will be tested with our main function.
- Before you submit your program, ask a peer to test your program and make sure it compiles.
- Since this program is the first part of a two-part assignment, you are required to complete it on time.