

# CSC1322 Advanced Programming in C++

## Programming Assignment: Monitoring Rhinos: Part II

### Objectives

- Learn to use functions you have already written (and tested).
- Learn to write boolean expressions that use short-circuit evaluation.
- Learn about the use of NULL and testing for it.

### Introduction

This assignment is a continuation of the previous one. Start with your correct rhino program you did previously and add additional functions to class Rhino and test them.

### What To Do

The steps to perform in doing this assignment:

**Step 1.** Add a print function. Write a print function with no parameters that produces a string representation of the Rhino object. The function header would be:

```
void Rhino::print()
```

Example output from the print function is given:

```
"Male rhino Juba. Born 5/1950. Tag 10. Mother Kibibbi. Children 1."  
"Female rhino Kibibbi. Born 1/1939. Children 2."  
"Male rhino Bubba. Born 11/1960. Tag 52. Mother Kibibbi.  
Father Jimma. Children 0."
```

Here are the rules:

- Avoid if-statement as much as you can. For better readability of your program, put each part on a separate line. You may use conditional expressions, and I give you one of them below.
- Exactly one blank appears between words (with the exception of the nickname, which can have any spacing within it, depending on what is in the nickname member variable). Periods '.' are as indicated in the examples.

- Get the gender with a blank after it using this **conditional expression**:

```
(isMale() ? "Male " : "Female ")
```

Note:

The meaning of a conditional expression is:

```
c = ((a == b) ? 2:3)
```

equivalent to

```
if(a == b)
```

```
    c = 2;
```

```
else
```

```
    c = 3;
```

- The word “rhino” is followed by a blank, the rhino’s nickname, and a period ‘.’.
- The birth month and year always appear, as in the examples.
- The tag is given in the form “Tag xxx. ” It appears only if the rhino has a tag.
- Suppose the mother is known and its nickname is “xxx”. Then “Mother xxx. ” appears as in the examples. Similarly for the father. If the mother (father) is not known, nothing appears about it. The mother appears first.
- The number of children appears as shown in the examples.

In testing the print function, you need enough test cases to ensure that each different way of evaluating a conditional expression is tested. For example, to test whether the gender appears correctly, you need at least two test cases, for a female rhino and for a male rhino.

**Step 2.** Comparison functions. Write the functions in the table below. Each produces a boolean value. Do one at a time.

Here are the ground rules for writing these functions:

- The names of your functions must match those listed exactly, including capitalization. The number of parameters and their order and types must also match. The best way to ensure this is to copy and paste.

Our testing program will expect those function names and parameters, so any mismatch will cause our testing program to fail. Parameter names are never tested, so you can change the parameter names if you want.

- Use logic relational operations (AND &&, OR || or NOT !) instead of if-else statement as much as you can.
- To receive full credit for isParentOf, it must be written in terms of calls to functions isMotherOf and isFatherOf. You need to learn to call functions already written instead of doing duplicate work, saving you time and avoiding redundant code. Writing code in this fashion is good software-engineering practice.

Functions (all return boolean values)
<p>✎ <code>isMotherOf(Rhino e)</code> “e is not NULL and this rhino is the mother of e”.</p> <p>Note: The notation above means precisely this: the value the function will return is true if e is not NULL and this rhino is e’s mother and is false otherwise. So, e may be NULL (in which case, the function will return false).</p> <p>The same comment holds for the remaining functions.</p> <p>✎ <code>isFatherOf(Rhino e)</code> “e is not NULL and this rhino is the father of e”.</p> <p>✎ <code>isParentOf(Rhino e)</code> “e is not NULL and this rhino is a parent of e”.</p> <p>✎ <code>isSisterOf(Rhino e)</code> “this rhino is e’s sister”. Precondition: e is not NULL.</p> <p>Note: B is A’s sister if (1) B and A are different rhinos, (2) B is female, and (3) B and A have at least one parent in common.</p> <p>✎ <code>isYounger(Rhino f)</code> “the function returns true if this object is younger than f, otherwise false.” Precondition: f is not NULL.</p>

**Step 3.** Testing. Write a client program (with a main function) to test each of the functions as well as Rhino class thoroughly.

## 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 all the three files into a single file and submit to the Blackboard. 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 second part of a two-part assignment, you should complete the first part.