

# Programming Languages

## Homework 2

**Due Wednesday, January 22nd at 2 AM**

1. Be sure your Java program from HW1 has base class Shape with one data member, name, of type string. Shape must have a constructor that takes a string parameter to initialize this name data member. Each derived class should have a constructor that takes a parameter for the name, plus parameters for any added data members, then call the base class (Shape) constructor with this name as a parameter, then initialize any data members added in the derived class. You should have a total of five classes in your Java program for Shape and its derived classes and another (sixth) class for Picture.
2. (100 points) Translate the entire class hierarchy you defined in HW1 into C as shown in lecture. Instead of using class Picture with a linked list of Shapes, you may use an array of Shapes in main as I did in my example linked below. As usual, start with [Wikipedia for an overview of virtual method tables](#). There are lots of images on the Internet that show how vtables, vpointers, and member function pointers are organized. [Here is one](#). Note that Brass::adjust is inherited from its parent class Wind. Most other methods are overridden (defined) in the class. Follow the following procedure for generating the definitions:

A) For each concrete class define the following:

- 1) a C struct for the data in each class instance plus the Vpointer
- 2) member functions
  - a) add `_this` formal parameter
  - b) prefix methods with `_this->`
- 3) vtable
- 4) constructor
  - a) call the parent constructor
  - b) set the vpointer

B) Write the main function

- 1) allocate an object of each type with malloc (and free them appropriately)
- 2) call the appropriate constructor on each object
- 3) use the formula for dynamic binding to call each method on each object

You can find my example files in the following two files along with a Makefile:

<http://www.ics.uci.edu/~klefstad/public/141/hw2/CircleTest.java>

<http://www.ics.uci.edu/~klefstad/public/141/hw2/Circle.cpp>

<http://www.ics.uci.edu/~klefstad/public/141/hw2/Makefile>

Note: I used C++ and the compiler g++, but only used C features. In C++, I implemented only class Circle and ACircle. There is a comment at the top of Circle.cpp that explains the algorithm for generating the definitions (similar to the one above). Focus on understanding how inheritance and polymorphism are implemented and avoid just copying what I have done.

**Output:**

Output should be similar to the previous assignment. I will provide your program with two arguments (arg1 and arg2) and you should use those as the dimensions for your objects. Similar to last assignment, be sure that your output follows this general structure:

FirstTriangle(5, 5) : 12.5  
SecondTriangle(4, 4) : 8

FirstCircle(5) : 78.54  
SecondCircle(4) : 50.27

FirstSquare(5) : 25  
SecondSquare(4) : 16

FirstRectangle(5, 5) : 25  
SecondRectangle(4, 4) : 16

Total : 231.31

You can put the actual shape drawings either all at the end (in any order), or immediately after you give the name and area of the shape. Ensure you follow the format for the shape name and area as above.

**Submit:** to Gradescope, submit one file containing all of your code under a file named ShapeTest.c. The autograder will be using the default installation of gcc version 7.4.0 to compile your code, so take this into account when writing your code.