

# CS 120 Project 4

Due on Blackboard by Friday, October 28

For this project you will design, implement, and test C++ class inheritance.

---

## Design

Think about what your parent and child classes will represent. What added methods does the child class have? What added fields does the child class have? Write either a diagram or bullet points outlining the answers to these questions.

Some examples:

- A Saxophone class that extends Musical\_Instrument
- A Sailboat class that extends Boat
- A First\_Person\_Shooter class that extends Video\_Game
- A HybridCar class that extends Car
- A Penguin class that extends Bird

---

## Implement

Your classes should be declared in header file(s) (with RMEs and other comments) and defined in corresponding .cpp file(s).

You may reuse a class from a previous project if appropriate.

---

## Test

To make sure your classes are working the way you designed, include a .cpp file that has the main function and tests both classes thoroughly.

---

## Grading

The project is out of 50 points.

### Design and Style

- 2 pts Are there at least three files and does each file have the correct code?
- 10 pts Are there sufficient comments and/or writings to explain what each method accomplishes and what each field represents? Are methods const when they should be?
- 4 pts Do the files follow the style guidelines from class? Are they readable? Do the names make sense?
- 10 pts Is there evidence of a well-thought-out design? Does each method have a clear purpose? Is this the best way to implement the class given the functionality goals?
- 4 pts Are the methods and fields members of the correct class? Is there a clear distinction between parent and child?

### Implementation

- 4 pts Are two classes implemented, one a subclass of the other?

4 pts Are keywords "virtual" and "override" used where appropriate?

#### Testing

4 pts Is every method tested (directly or indirectly)?

8 pts Does testing cover all possible cases?

---

### Extra Credit

To earn extra credit, you must complete the following:

1. Create an abstract class that has at least two subclasses.
2. The subclasses must override the pure virtual method(s) in the parent class.
3. You must create a vector of pointers to the abstract class.
4. The vector must contain pointers to child class objects (a mixture of the 2+ subclasses).
5. You must demonstrate polymorphism by looping through the vector and calling the same method on each element (at least one method).

If you complete all five steps, you will earn 20 points of extra credit. This is an all-or-nothing option, no partial credit. If you miss a step, you do not earn extra credit.