Person<-Student

Inheritance introduces quite a few new possibilities into your programs. It is easy to miss some of the details that you really **must** master to make effective use of the object-oriented technique of inheritance.

So, instead of working with fun stuff, like card games and shooting down aliens, we'll start by returning to the old, boring "finger-exercise" example that lets you concentrate on one piece of the inheritance puzzle at a time.

Click the Running Man to open the lab example in **Replit**. **Fork** the **Repl** so you'll have your own copy to work on.

Extending Person

In Java, you use the `extends` keyword to specify the parent or **superclass** (called the **base class** in C++) when you define the child or **subclass** (called the **derived class** in C++). Instead of using the `extends` keyword, as in Java, we **use a colon** in exactly the same position. In addition, we specify that the **base class** is `public`.

```
class Student : public Person
{
    // members of the derived class
};
```

`Student` is the derived class, while `Person` is the base class. Each of these class definitions is placed in its own header file, with the implementation of the member functions in `person.cpp` and `student.cpp`. You can open these in the Replit editor.