References & Pointers

While slicing is a problem it is not the only culprit here. Even without slicing, the code would still not work because in C++ **polymorphism only works with references or pointers**.

To see, this, make the following changes to **`main.cpp`**:

```
Student sam = Student("Sam", 201795);
Person pam = Person("Pam B.");
Person& samRef = sam;
Person* samPtr = &sam;
cout << "sam says->" << samRef.toString() << endl;
cout << "sam says->" << samPtr->toString() << endl;
cout << "pam says->" << pam.toString() << endl;
```

Now, the **`Person&`** reference **`samRef`** refers to the Student object **`sam`**, and when we call **`samRef.toString()`** it calls **`Student::toString()`**, not **`Person::toString()`** like our previous examples did.

The same thing happens if we use the **`Person* samPtr`**. It is also polymorphic.