Pointers to the Rescue

One solution is to create a `vector<Person*> v` or an array `Person* a[2]`. Here's a short example that places two different kinds of **`Person`** pointers in a **`vector`** and prints them. Each person responds appropriately. Go ahead and add this code to **`main()`**. Include the **`<vector>`** header.

```
int main()
{
    vector<Person*> people;
    people.push_back(new Student("Sam", 201795));
    people.push_back(new Person("Pam B."));

    for (auto p : people) {cout << p->toString() << endl;}
    for (auto p : people) delete p;
}
```

Since two of these objects are created on the heap, it is up to you to reclaim their memory before the **`vector`** goes out of scope and it is lost. The **`vector`** cannot do it because it does not know if the pointers it contains point to objects on the heap or objects on the stack. If you add a stack-based pointer to this program, it crashes.