Computer Information Systems 23 - Homework 3

Cody Vig

Problem 1

Consider the class `studentType`:

```
class sutdentType: public personType
{
public:
    void print();
    void calculateGPA();
    void setID(long id);
    void setCourses(const string c[]);
    void setGrades(const char cG[], int noOfC);

    void getID();
    void getCourses(string c[], int noOfC);
    studentType(
        string fNmae = "", string lName = "", long id = -1,
        sting *c = nullptr, char *cG = nullptr, int noOfC = 0
    )
}
```

1. Is this a concrete class?

As writtes, yes, `studentType` is a concrete class. It does not have virtual functions, and as such there are no issues with instantiation.

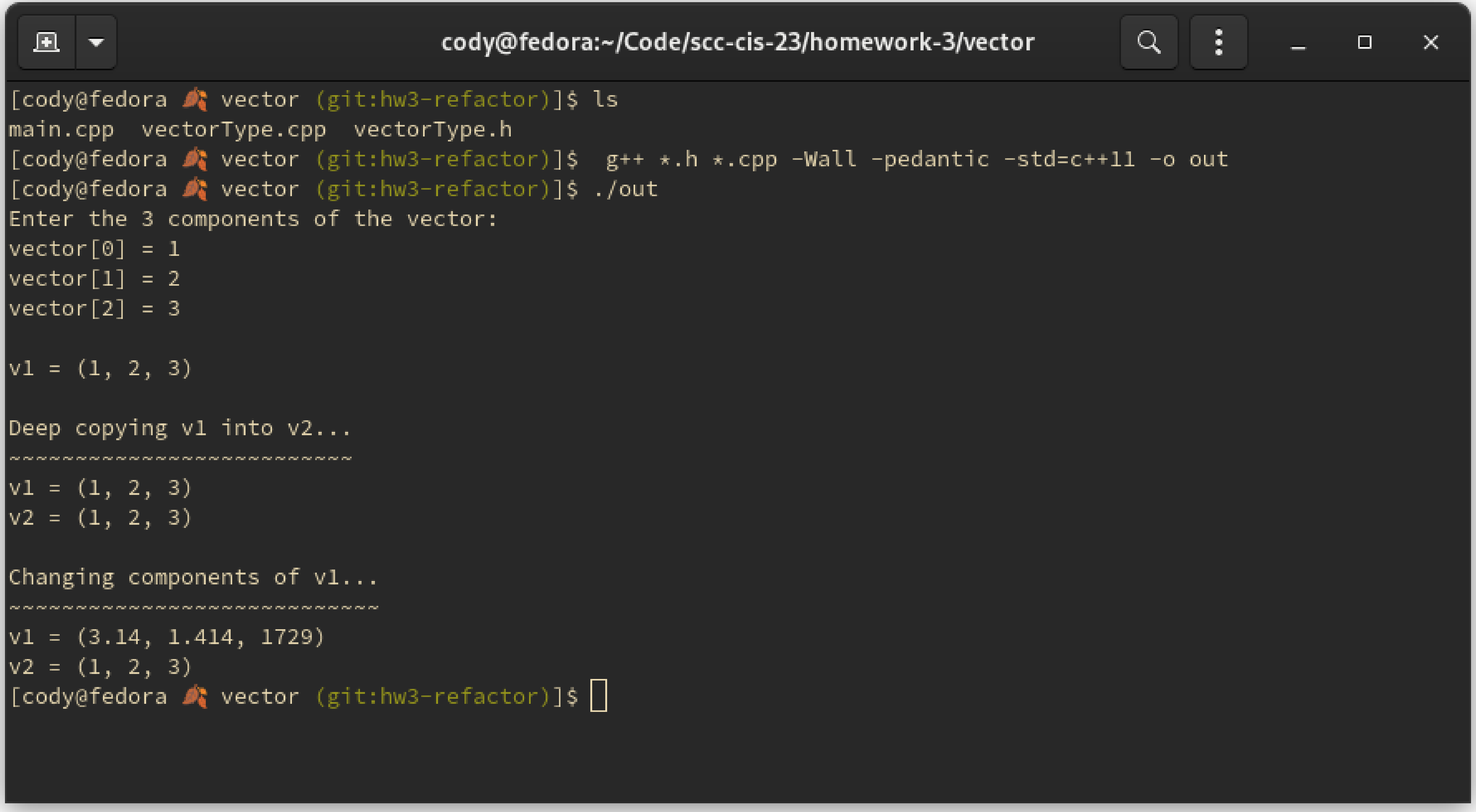
2. How would you change the definition of the class student so that the functions print and calculateGPA are pure virtual functions?

To make them virtual functions, we would need to add the `virtaul` modifier to their prototypes. To make them *pure* virtual functions, we need to remove all declarations of those functions and end their prototypes with `= 0`:

```
virtual void print() = 0;
virtual void calculateGPA() = 0;
```

3. With this change, is studentType a concrete class? Can you create studentType objects?

By definition, the existence of at least one pure virtual function makes this class an abstract class, not a concrete one. As such, we cannot instantiate `studentType`, so we cannot create `studentType` objects.



```
[cody@fedora 🍂 vector (git:hw3-refactor)]$ ls
main.cpp  vectorType.cpp  vectorType.h
[cody@fedora 🍂 vector (git:hw3-refactor)]$  g++ *.h *.cpp -Wall -pedantic -std=c++11 -o out
[cody@fedora 🍂 vector (git:hw3-refactor)]$ ./out
Enter the 3 components of the vector:
vector[0] = 1
vector[1] = 2
vector[2] = 3

v1 = (1, 2, 3)

Deep copying v1 into v2...
~~~~~~~~~~~~~~~~~~~~~~~~~~
v1 = (1, 2, 3)
v2 = (1, 2, 3)

Changing components of v1...
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
v1 = (3.14, 1.414, 1729)
v2 = (1, 2, 3)
[cody@fedora 🍂 vector (git:hw3-refactor)]$
```