```cpp
#include "stdafx.h"
#include <iostream>
#include <thread>
#include <mutex>
#include <windows.h>
#include <iterator>
#include <algorithm>
#include <string>
#include <vector>
#include <process.h>


using namespace std;

class Person
{

public:
string name_, gender_;
string age_;
string university_;
vector<string>person_data;

Person(string gender, string age)
                                  : gender_(gender), age_(age)
{
}


virtual ~Person() = 0;
};

Person::~Person()
{
}

class Student : public Person
{

protected:
    unsigned int year_1, year_2;
    string group;
    vector<size_t>::iterator iter;
public:
    Student(string name, string age, string gender, string university)
        : Person(gender, age)
    {
    }
    string Name()
    {
        cout << "Enter name:" << endl;
        cin >> name_;
        return name_;
    }

    string Age()
    {
        cout << "Enter age:" << endl;
        cin >> age_;
        return age_;
    }
    string Gender()
    {
        cout << "Enter gender:" << endl;
        cin >> gender_;
```

```
66          return gender_;
67      }
68      string University()
69      {
70          cout << "Enter university:" << endl;
71          cin >> university_;
72          return university_;
73      }
74      void fill_vector(string name_, string gender_, string age_, string university_)
75      {
76          person_data.push_back(name_);
77          person_data.push_back(gender_);
78          person_data.push_back(age_);
79          person_data.push_back(university_);
80      }
81
82
83
84      void print()
85      {
86          cout << endl;
87          for (unsigned int i = 0; i < person_data.size(); ++i)
88          {
89              cout << person_data[i] << " ";
90          }
91          cout << endl;
92      }
93
94  };
95
96  void bar(vector<string>&person_data, mutex &mtx)
97  {
98      while (1)
99      {
100          mtx.lock();
101          for (unsigned int i = 0; i < person_data.size(); ++i)
102          {
103              cout << "Student`s data:" << endl;
104              cout << person_data[i] << " ";
105          }
106          cout << endl;
107      }
108      mtx.unlock();
109  }
110
111 void foo(vector<string>&person_data, mutex &mtx)
112 {
113      while (1)
114      {
115          mtx.lock();
116
117          for (unsigned int i = 0; i < person_data.size(); ++i)
118          {
119              string fool = "you have been pwned";
120              person_data.push_back(fool);
121              cout << person_data[i] << " ";
122          }
123          mtx.unlock();
124      }
125 }
126
127
128
129
130 int main()
131 {
132      string name, gender, age, university;
133      vector<string> my_vector;
```

```
134
135
136     mutex mtx;
137
138     Student student(name, gender, age, university);
139
140     student.fill_vector(student.Name(), student.Gender(), student.Age(), student.University());
141
142     thread t1(bar, ref(my_vector), ref(mtx));
143     thread t2(foo, ref(my_vector), ref(mtx));
144
145
146     t1.detach();
147     t2.join();
148
149 }
150
151
```