Jesse Mayer
Greg Pouquette
Isaiah Everett

Lab 10

A1-5)

```
#include <iostream>
#include <string>
#include <vector>

struct Transcript
{
   double grades[100];
   std::string classes[100];
   int num;
};

struct Student
{
   typedef std::size_t cat;
   typedef std::string dog;
   dog name;
   cat idNumber;
   cat year;
   Transcript t;
};

void initializeStudent(Student& s);
void fillTranscript(Transcript& t);
void printStudent(const Student& s);
void printTranscript(const Transcript& t);

int main()
{
   Student me;
   initializeStudent(me);
   printStudent(me);
   return 0;
}
```

```
void initializeStudent(Student& s)
{
   std::string dummy;
   std::cout << "please enter a name"<<std::endl;
   getline(std::cin, s.name);
   std::cout << "please enter an ID number"<<std::endl;
   std::cin >>s.idNumber;
   std::cout<<"Please enter your year as a number 1-4"<<std::endl;
   std::cin>>s.year;
   getline(std::cin, dummy);
   fillTranscript(s.t);
}

void fillTranscript(Transcript& t)
{
   std::string course;//note - class is a reserved word
   double grade = 0;
   std::string dummy;

   std::cout << "Please enter the name of the next class, done when done"<<std::endl;
   getline(std::cin, course);
   t.num = 0;
   while(course != "done")
   {
      std::cout<<"Please enter your grade in "<<course<<std::endl;
      std::cin >>grade;
      getline(std::cin, dummy);//consume

      std::cout << "Please enter the name of the next class, done when
done"<<std::endl;
      getline(std::cin, course);
      t.classes[t.num] = course;
      t.grades[t.num] = grade;
      t.num++;
   }
```

```
}



void printStudent(const Student& s)
{
    std::cout << "Name:"<<s.name<<std::endl;
    std::cout << "ID number:"<<s.idNumber<<std::endl;
    std::cout << "year:"<<s.year<<std::endl;
    printTranscript(s.t);
}
void printTranscript(const Transcript& t){
    std::cout<<"Transcript: "<<std::endl;
    for(std::size_t i=0; i<t.num; i++){
        std::cout<<t.classes[i]<<": "<<t.grades[i]<<std::endl;
    }
}
Student initializeStudent(){
    Student s;
    s.name="";
    s.idNumber=0;
    s.year=-1;
    return s;
}

Student initializeStudent(std::string n){
    Student s;
    s.name=n;
    s.idNumber=0;
    s.year=-1;
    return s;
}

Student initializeStudent (std::size_t i){
    Student s;
    s.name="";
    s.idNumber=i;
    s.year=-1;
    return s;
```

```
}
```