

Intro to Computing

Lab 13

November 18, 2009

Objectives: You will gain experience using C++:

- C++ string class: constructors and member functions
- C structures: definition, usage, arrays of structures, and passing as parameters

Download the following file to your desktop: <http://www.cs.uni.edu/~fienup/cs051f09/labs/lab13.zip>

Extract this file by right-clicking on lab13.zip icon and selecting Extract All.

Part A: Yesterday in class we started implementing the `splitLineByCharacter` function which takes as input a string (`myLine`) and a character delimiter, and returns an array substrings and `substringCount`. For example if `myLine` contained:

		Position in the string				
Inputs:		0123 4567890 123	myLine: <input type="text" value="Mark\tFienup\t84"/>	delimiter: <input type="text" value="\t"/>		
Outputs:		0	1	2	3	
substrings: <input type="text" value="Mark"/> <input type="text" value="Fienup"/> <input type="text" value="84"/> <input type="text" value=""/> <input type="text" value=""/>					substringCount: <input type="text" value="3"/>	

The lab13.zip file you downloaded and extracted contains a `PartA` folder with a Visual Studio C++ project file: `testStringClass.sln` inside. Double-click on it to open this project in Visual Studio. Complete the `splitLineByCharacter` function we started yesterday in lecture.

After you have debugged your `splitLineByCharacter` function, raise your hand and demonstrate your program.

Part B: A *structure* is a C (and C++) construct that allows multiple variables of potentially different types to be grouped together. The general format for defining a structure is:

```
struct <structName>
{
    type1 field1;
    type2 field2;
    ...
}; // NOTE the ';' after the definition
```

For example, we can define the template for a `Student` structure as:

```
struct Student {
    int studentID;
    string name;
    short yearInSchool;
    double gpa;
}; // end Student struct
```

Like the recipe for a cake, we don't actually have any `Student` structures until we define variables by using the structure-name `Student` as a type name:

```
Student bill = {123456, "Bill", 3, 3.10}; // initialize with values
Student sally, myClass[50]; // empty members
```

bill:	studentID <input type="text" value="123456"/>
	name <input type="text" value="Bill"/>
	yearInSchool <input type="text" value="3"/>
	gpa <input type="text" value="3.1"/>

sally:	studentID <input type="text"/>
	name <input type="text"/>
	yearInSchool <input type="text"/>
	gpa <input type="text"/>

The dot (.) operator is used to refer to members of struct variables:

```
cout << "Enter sally's student ID: ";
cin >> sally.studentID;
cin.ignore(50, '\n');

cout << "Enter sally's name: ";
getline(cin, sally.name);

sally.yearInSchool = 2;
sally.gpa = 3.75;
myClass[0].gpa = 4.0;
```

The lab13.zip file you downloaded and extracted contains a PartB folder with a Visual Studio C++ project file: PartB.sln inside. Double-click on it to open this project in Visual Studio.

- a) Write a function displayStudent to display the information about a student to the console. Call it from the main program using the variable bill.
- b) Unlike arrays, a structure variable definition does NOT create a pointer to a chunk of memory. What implications done that have for passing structures as parameters? Write a function getStudent to interactively read information about a student from the keyboard. Call getStudent from the main program using element 25 of the array myClass, then call displayStudent to display the information about this student.
- c) The file classInfo.txt contains student information about students in a class. The first line in the file contains a count of the number of students. The remaining lines contain the information about the students. Each student's information is spread across 4 lines with the first line containing the ID, the second line contains their name, the third has their year-in-school code, and finally the fourth line has their GPA. Write two functions:
 - getMyClass - to read the file classInfo.txt into the myClass array and update a count of the number of students in the array
 - displayClass to display all the students to the console by calling displayStudent repeatedly

After you have completed and tested the above modifications, raise your hand demonstrate your program.

If you do not get done today, you can show me the completed lab in next week's lab period.