Recitation 1

Opening a file

Part of this recitation will focus on teaching fstream and sstream, two of the iostream features in C++. The fstream library contains functionality to read and write to files. We will use the ifstream, which is used to read data from a file. We will also use the stringstream features included in the sstream library, which make it possible to treat a string as a stream of characters that our program can process. The value of stringstream in this recitation is that we can read data from the file into a string, and then use a delimiter to pull out the individual values in the string.

In the following code, we open a file and then use stringstream to parse each line in the file. In your program, make sure to include the necessary libraries at the top of your program.

```
#include <iostream>
#include <fstream>
#include <sstream>
using namespace std;
int main() {
  ifstream inFile;
  // create an instance, can be named whatever you want
  string data;
  inFile.open("filename.txt"); //open the file
  if(inFile.good()){ //error check
    cout<<"opened successfully"<<endl;
    while(getline(inFile, data)){ //read/get every line of the file & store it
      cout<<data<<endl; //can see the data (each line) printed
      stringstream ss(data); //create a stringstream variable from string data
      int elementOne; ss>>elementOne;
      cout<<elementOne<<endl;
      string elementTwo; ss>>elementTwo;
      cout<<elementTwo<<endl; }
  }
  else{
    cout << "File unsuccessfully opened" << endl; }
  inFile.close(); //close the file
  return 0;
}
```

If this file had the information:
12 13
then the data would print:
12
13

Structs

A struct is a collection of data elements grouped together to create a composite data type. They can be made up of different types, including other structs, which is what makes them useful.

```
//Creating a struct data type: template to define all instances
struct CarData{
  string model;
  string make;
  string year;
}
```

In the above example, CarData is the type of struct we chose. The make, model, and year are members of this struct. Every instance of CarData created will contain all three members.

```
int main(){
  CarData cd; //create an instance of CarData called cd
}
```

Above is how to create an instance of CarData. This is the same as a normal data type we have seen, such as int, string, or double. After the instance is created, we can access the members of the instance using the dot notation, such as

cd.make or cd.model

to access the make and model members of cd.

Array of Structs

Structs can also be used as the type of an array, just like any other variable type. In this example, we create an array, called carArray, of CarData with a fixed size of 10:

```
CarData carArray[10]; //similar to creating array of ints

for(int i=0; i<10;i++){
  cout<<"make:"<<caryArray[i].make<<" model:"<<carArray[i].model<<
  " year:"<<carArray[i].year<<endl;
}
```

Recitation Programming Exercise

For Recitation 2, you will be writing a program that opens the provided text file of Car information, located on moodle. Starting with the above struct and the appropriate libraries, you should store that files information in an array of structs. Your program also needs to check if the file opened correctly before reading to the end of the file. Once the data is stored in an array, your program should print the contents of the array from beginning to end. These concepts have been discussed above or in the lecture notes. You should use Google for further resources.

How to Submit

When youre done with your program, show your work to your TA and submit your Recitation2.cpp file to the Recitation 2 link on Moodle.