Structs and Classes

10 17 2001

Opening Discussion

- What did we talk about last class?
- When you want to operate on every element of a two dimensional array, you will typically need to have a doubly nested loop.
- From the reading in chapter 7, who can tell me what a struct is? What about a class?

Grouping Data struct

- It is often useful to be able to group together separate data elements under a single type and name. In C/C++ this is done with a struct.
- Using structs gives us the ability to do things like make a Student that has two strings and a double in it instead of keeping multiple parallel arrays.

```c++
struct Student {
    string first, last;
    double grade;
};
```
New Types

When you define a struct (or a class) you are creating a new type. You can declare variables of that type and pass them in exactly the same way you would the primitive types that we have discussed previously.

```c
Student student1, student2;
Student PAD[25];
```

Using structs

- Accessing elements
  ```c
  Student1.last="Lewis";
  cin >> PAD[i].grade;
  ```
- Passing as arguments of a function. You should typically pass by reference for efficiency. Use `const` if appropriate.
  ```c
  void readStudent(Student &s);
  void printStudent(const Student &s);
  ```

Grouping Data and Functions class

- Objects group both data and functions to operate on them into a single unit called a class. In a class they are called by different terms: members/properties and methods.
- When a method is invoked for an object it automatically has access to all of the members and methods of that object as if it had been passed as an argument.
**Public, Protected, Private**

- Members and methods of a class are also given a level of visibility. These are specified by public, protected, and private.
- All functions have access to the public members and methods.
- Only the methods of that class have access to private members and methods. (Friend classes can also see them.)
- Protected members and methods are like private ones but can also be seen by subclasses.

**Interface vs. Implementation**

- The ability to hide parts of a class by declaring them private is a major benefit of OOP. The public parts of a class are often called the interface. What goes on to make the interface work is the implementation.
- Good object-oriented designs have all the members private so the implementation is completely hidden.

**Minute Essay**