

CLASSES

What do Classes Do?

- Classes bind several items together into a single variable.
- Data items can be different types.
- The pieces (Fields or Members) are identified by name.
- Can also associate functions with the type.

A Simple Class – No Functions

- A class for a telephone book entry might look like:

```
class TelephoneEntry {
public:
    string name;
    int areaCode;
    int number;
}; // Semicolon is important!!!
```

The Pieces

- The word "class" at the top tells the computer that the definition of a class follows. The name can be any valid identifier.
- Public tells the computer that anyone can access the field/member.

Declaring A Variable of the Class Type

- Example

```
int main (void)
{
    TelephoneEntry myphone;

    myphone.name = "Jerry Heuring";
    myphone.areaCode = 419;
    myphone.number = 5308196;
```

More lines....

```
cout << myphone.name << " "
     << "(" << myphone.areaCode
     << ")" << myphone.number/10000
     << "-"
     << myphone.number % 10000
     << endl;
return EXIT_SUCCESS;
}
```

Why Classes

- Can treat an entire group of data as one object.
- Can hide details (Why would you want to hide details?)
- Can restrict access and the way the item is used.

Other Features

- Members of the class can be either public, private, or protected.
- Public members are accessible by everybody.
- Private members are only accessible from functions inside the class
- Protected members are accessible from the functions in the class and from other classes "derived" from the class.

Adding a Print Routine

```
class TelephoneEntry {  
public:  
    string name;  
    int areaCode;  
    int number;  
    void print() { // Adds a print function  
        cout << name << " "  
            << "(" << areaCode  
            << ")" << number / 10000  
            << "-" << number % 10000  
            << endl;  
    }  
}; // Semicolon is important!!!
```

Some Notes on Style

- Usually don't want to leave data public.
- Often have access routines to get data.
- Appear at the start of a program (before the main routine or other functions).

The Importance of Classes

- Classes provide a nice way to "encapsulate" the functionality of an object. If done properly the user of the class doesn't need to know why/how it works.
- Basis for Object-Oriented programming in C++
- Already seen some in ifstream and ofstream as well as string.

A Class With a Constructor

- Many classes have "Constructors" that are used to initialize objects or instances of the class when they are created.
- A constructor is a member function with the same name as the class.

Telephone Entry

```
class TelephoneEntry {  
public:  
    string name;  
    int areaCode;  
    int number;  
    TelephoneEntry ( string n, int c, int nbr)  
    {  
        name = n;  
        areaCode = a;  
        number = nbr;  
    }  
}; // Semicolon is important!!!
```

Again

```
class TelephoneEntry {  
public:  
    string name;  
    int areaCode;  
    int number;  
    TelephoneEntry ()  
    {  
        name = "";  
        areaCode = 0;  
        number = 0;  
    }  
}; // Semicolon is important!!!
```