Strings and Collections

Kate Gregory

www.gregcons.com/kateblog @gregcons

Objects and Classes

C++ apps are not just made of functions, but of classes and objects too

C++ is an object oriented language

Objects and Classes

A class defines the idea of an object

What data it holds

What functions it can be asked to perform

Example: Date

Objects and Classes

An object is an instance of a class

- Example: May 1st, 1990 or Dec 3rd, 2017

Functions inside a class are called *member functions*

The kind of functions shown earlier are called *free functions* or *nonmember functions*

C++ uses plenty of both

Strings

```
#include <string>
```

C++ has a very useful string class in the `std` namespace

- #include <string>

Can compare, combine and manipulate strings

Also search for substrings, make replacements, ...

Makes string feel like a built in type

For Unicode, use `wstring`

String Operators

To combine two strings: + +=

To test two strings: == < > !=

The cout << operator and cin >> operator both work perfectly with strings

Reuse Your Knowledge

You know how to compare two integers with > and similar operators

- Compare strings the same way

You know how to add two integers with +

- Add strings the same way

You know how to print an integer on the console with cout >>

- Print strings the same way

The more you know how to do, the easier new things are to learn

String Member Functions

```
string greeting = "Hello, ";

int len = greeting.length();


string s2 = greeting.substr(2,3);


int pos = greeting.find(“He”);
```

◀ length

◀ substr

◀ find

Exercise

Write a program that asks the user for two words and tells them which is longer

Hints:

- Use the code from Guess My Number as a starting point
- This app can run until the user says to stop, or just once: your choice

Once it's working, try entering a phrase and see what happens

Collections

Many programs need to work with a number of similar items

- The people in a department
- The items in an order
- The transactions in an account

The Standard Library provides classes that are ready to use

Collections

Simplest and best first choice:
`vector`

Holds a number of values, all of the same type

Size does not need to be known in advance

Easy to access a specific item or all of them

```
#include <vector>
using std::vector

vector<int> nums;

nums.push_back(3);

for(auto item:nums)

{

        cout << item << " ";

}

nums[0]=7;
```

- Include the header file and simplify the name
- When declaring, specify what it is a vector of
- push_back adds an element at the end
  - type must match
- The ranged for loop uses each element in the vector in order
- Access elements with []
  - Zero based

```
#include <algorithm>


vector<string> words;

// . . .

sort(begin(words),end(words));


int threes =
   count(begin(nums),end(nums),3);
```

- Free functions that work with vector and other collection classes
- Begin and end are here because you can actually sort just part of a vector if you wanted to
- The count() function looks for values matching the third parameter

Operator Overloading

Operators are just functions

strange names, no ()

You've seen many operators in this module

string

+, +=, ==

vector

[]

cout and cin

>> and <<

Templates

Templates are a powerful way to write a library

- Work on any type, without giving up type safety
- Work on both built in and user defined types
    - int, bool, double, string, Employee, OrderItem, ...
    - Operator overloads are a big part of that

Using a template is not difficult

- vector
- sort()

Summary

The string class is powerful and useful

- Intuitive operator overloads
- Member functions
- Works with some free functions in the standard library as well

The Standard Library includes classes to represent a collection of anything

- vector is the most generally useful collection
- There are free functions to work with vector and other collections

The template mechanism in C++ allows us to generalize over types without losing type safety

- You write less code
- Programs have less bugs

Operator overloading is extremely powerful