Azure

CS196

Microsoft Azure Workshop

Getting Started with Microsoft Azure
Hands on Tutorial

Swag, including a raffle for a Razer bundle worth $120 and including a mechanical keyboard, gaming mouse, gaming mouse pad, and overear headphones

7pm TONIGHT! (2/27)
Siebel 0216

Microsoft

C++ File Structure and Intro to the STL

G Carl Evans

C Variable Definitions

```
int i = 0;
double radius = 6.5;
const double kPi = 3.14159;
int students_per_lecture[] = {155,121};
char *program_name = argv[0];

std::vector<double> grade_cutoffs = {0.9,0.8,0.7};
Monster cs126_prof;
Room dcl_1320;
```

C++ function definitions and declarations

Function Declarations

```
int TimesTwo(int x);
```

Function Definitions

```
int TimesTwo(int x) {
     return x * 2;
}
```

C++ File Stucture

Two file types

- `.cpp`
  - This is the file type that contains the definitions and code.
- `.h`
  - This is the file type that contains declarations.

Using header files

```
#include <iostream>
#include <vector>
#include <stdlib.h>
#include "libmult.h"
```

Writing Header Files

```
#ifndef LIBMULT_H
#define LIBMULT_H

//declarations here

#endif
```

C++ Bool and Bool Experssion

- **`bool`**
  - fundamental type this will hold either **`true`** or **`false`**

- **`true`**
  - This has the value of 1
- **`false`**
  - This has the value of 0

- A nonzero value will be coerced to **`true`**
- A zero value will be coerced to **`false`**

Standard Loops in C++

```
while(boolexpr){
  body
}

for(intitialize; boolexpr; reloop){
  body
}

for(range_decl : range_expr){
  body
}

do{
  body
}while(boolexpr)
```

Standard Branching

```
if(boolexpr) {

} else if (boolexpr) {

} else {

}
```

C++ Strings

- C – strings
  - C++ being based on C supports C style strings using **`char *`**
  - A pointer to an array of bytes terminated by a null byte

- <string>
  - This class provides a modern **`string`** interface to replace the C style stings
  - **`string1 == string2`** compares strings not pointers via overloading
  - Supports modification using methods including **`insert`**, **`erase`**, and **`push_back`**
  - Supports STL interfaces

C++ Libraries – I/O

- <iostream>
  - `cout << "thing to print";`
    - Prints all the fundamental types
    - Prints **`string`** variables
    - Can be extended to print other types
  - `cin >> variable;`
    - Reads input from standard input
    - Parses all fundamental types
    - Spaces separate different inputs
    - Will read words into **`string`** variables
    - White space separates inputs and input is read with enter

C++ STL

- The Standard Template Library
  - C++ library parameterized by types similar to Java generics
  - Containers
  - Iterators
  - Algorithms
  - Functions

C++ Libraries – Variable Length Arrays

- <vector>

C++ Libraries – Dictionary or Associative Array

- <map> or <unorderd_map>