

Function Usage

Lecture 14

Sections 6.3, 6.4

Robb T. Koether

Hampden-Sydney College

Mon, Sep 30, 2013

1 Function and Parameters

2 The Function Interface

3 Header Files

4 Assignment

Outline

- 1 Function and Parameters
- 2 The Function Interface
- 3 Header Files
- 4 Assignment

Functions and Parameters

- A **function** is like a program within a program.
- Every function has a **name**.
 - `rand()`
 - `sqrt()`
 - `pow()`
- A function may have any number of **parameters**, including none, always written within parentheses.
 - `rand()`
 - `sqrt(x)`
 - `pow(x, 5)`

Return Values

Return Values

```
double r = sqrt(x);  
int n = floor(y);  
srand(time(0));
```

- A function may **return** a value of a specified type.
- If it does not return a value, then it is a **void** function.

Function Usage

Function Usage

```
b = a * sin(angleB) / sin(angleA);  
srand(time(0));  
c = sqrt(pow(a, 2) + pow(b, 2));
```

- A function reference may be used anywhere that an object of its return type is permitted.
- If a function is **void**, then it must be used on a line by itself.
- The parameters of a function call may themselves be function calls.

Outline

1 Function and Parameters

2 The Function Interface

3 Header Files

4 Assignment

Function Interfaces

- A function **interface** contains
 - Exactly the information needed by the programmer in order to use the function correctly.
 - Exactly the information needed by the compiler in order to check the usage of the function.

Function Interfaces

- The function interface specifies
 - The function name
 - The return type
 - The number of parameters
 - The types of the parameters
 - The order of the parameters
 - Which parameters are constant (if any)
 - The method of parameter passing (value or reference)
 - Whether the return value is constant
 - The method of passing back the return value (value or reference)
 - Which parameters have default values (if any)
- It does not need to specify the names of the parameters.

Function Prototypes

```
int rand(void);  
double sqrt(double);  
double pow(double, double);  
double pow(double, int);  
void srand(unsigned int);
```

- A function **prototype** is C++'s method of specifying a function interface.
- The same function name may have several different prototypes; they represent different functions of the same name.
- Visit the web site www.cplusplus.com/reference for more details.

Outline

1 Function and Parameters

2 The Function Interface

3 Header Files

4 Assignment

Function Prototypes

- The function prototype must appear before the function is used.
- Typically, the function prototypes are put in a **header** file and included using `#include`.
- The benefit of this is that we may write the prototype only once (in the header file) and then include it in as many programs as we wish.

Example: Header Files

- Example

- `HeaderFileExample.cpp`

Outline

1 Function and Parameters

2 The Function Interface

3 Header Files

4 Assignment

Assignment

Assignment

- Read Sections 6.3, 6.4.