Function Usage

Lecture 15
Sections 6.3, 6.4

Robb T. Koether

Hampden-Sydney College

Mon, Sep 30, 2019

Outline

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() // No parameters`
  - `sqrt(x) // One parameter`
  - `pow(x, 5) // Two parameters`

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);  // Law of Sines
c = sqrt(pow(a, 2) + pow(b, 2));    // Pythagorean Thm
```

- A function reference may be used anywhere that an object of its return type is permitted.
- The parameters of a function call may themselves be function calls.

Function Usage

Function Usage

```
srand(static_cast<unsigned int>(time(0)));
```

- If a function is **`void`**, then it must be used on a line by itself.

Outline

Function Interfaces

- A function interface contains
  - Exactly the information needed by the programmer in order to use the function correctly,

  which is also (almost)
  - Exactly the information needed by the compiler in order to check the usage of the function.

Function Interfaces

- A function interface contains
  - Exactly the information needed by the programmer in order to use the function correctly,

  which is also (almost)
  - Exactly the information needed by the compiler in order to check the usage of the function.
- This is called the function's prototype.

Function Interfaces

- The function prototype 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 (by value or reference)
  - Whether the return value is constant
  - The method of passing back the return value (by 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);
void srand(unsigned int);
```

- Visit the web site `www.cplusplus.com/reference` for more details.

Function Prototypes

```
double pow(double, double);
double pow(double, int);
double pow(int, double);
int pow(int, int);
```

- 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, especially the `ostream` class and **`operator<<`**.

Outline

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

Assignment

Assignment

- Read Sections 6.3, 6.4.