

Module 22: Function Prototypes

Intro to Computer Science 1 - C++
Professor Scott Frees

Textbook

This topic is covered in Section 6.8 in the text

Function Prototypes

- Some of our programs will have many functions
 - Function may call other functions...
 - Functions cannot be used before they are defined.
- Can “declare” functions before actually “defining how they work” (code).

Function Prototypes

```
double myfunction1(int parameter);  
double myfunction2(int parameter);
```

function prototypes

```
int main() {  
    cout << myfunction1(5) << endl;  
    cout << myfunction2(5) << endl;  
}
```

```
double myfunction1(int parameter) {  
    return parameter + 1;  
}  
double myfunction2(int parameter) {  
    return parameter - 1;  
}
```

function definitions

Guidelines

You may omit parameter ***names*** from function prototypes

- Instead of `int sum (int x, int y)`
- You may write `int sum (int, int)`

However - this doesn't mean you can leave out parameter names in the definition (code)

- This wouldn't make any sense!

Guidelines

Your prototypes **must** match up with definitions

- If you change your return type, name, or parameter types, you must remember to make the associated changes in the prototype (or vice-versa)
- Failure to do so will result in *linking* errors (at a minimum)

Remember, prototypes are *promises* to the compiler that you will eventually define the full function...

Programming Example 26

The constant “e” is found in many mathematical equations. Its value is approximately 2.718 and has been calculated to 869,894,101 decimal places.

The value of e^x can be approximated by the following (infinite) series. The larger “n” is, the more precise the approximation.

$$e^x = x^0/0! + x^1/1! + x^2/2! + x^3/3! + x^4/4! + \dots + x^n/n!$$

Allow the user to enter n - print out the estimate