

楓橋夜泊

~ 張繼

月落烏啼霜滿天，
江楓漁火對愁眠。
姑蘇城外寒山寺，
夜半鐘聲到客船。



Chapter 5



Functions

Examples of Functions

□ $y = f(x) = 2x^3 - 3x^2 - 3x + 2$

□ $f(x) = \sin(x) / \cos(x)$

□ $g(x) = \sin(x) / x$

Why Do You Need Functions

- ❑ A function is a self-contained **block of code** with a specific purpose.
- ❑ Sometimes you have to **repeat** the same task several times in a program.
 - With functions, you don't need to replicate the same code at various points.
- ❑ **Decompose** a large program into smaller functions makes it easily manageable for development and testing.
 - A typical program will consist of a large number of small functions, rather than a small number of large functions.

Example of Repeated Tasks (1)

```
#include <iostream>
using std::endl;
using std::cout;

int main()
{
    for (int j=1; j<=5; j++)
    {
        for (int i=0; i<j; i++)
            cout << '*';
        cout << endl;
    }
    for (int j=5; j>=1; j--)
    {
        for (int i=0; i<j; i++)
            cout << '*';
        cout << endl;
    }
    return 0;
}
```

```
*
**
***
****
*****
*****
****
***
**
*
```

Example of Repeated Tasks (2)

```
#include <iostream>
using std::endl;
using std::cout;

int print_stars(int n)
{
    for (int i=0; i<n; i++)
        cout << '*';
    cout << endl;
    return 0;
}

int main()
{
    for (int j=1; j<=5; j++)
        print_stars(j);
    return 0;
}
```

*
**

Example of Repeated Tasks (3)

```
#include <iostream>
using std::endl;
using std::cout;

int print_stars(int n)
{
    for (int i=0; i<n; i++)
        cout << '*';
    cout << endl;
    return 0;
}

int main()
{
    for (int j=1; j<=5; j++)
        print_stars(j);
    for (int j=5; j>=1; j--)
        print_stars(j);
    return 0;
}
```

```
*
**
***
****
*****
*****
****
***
**
*
```

Example: Coin Tossing

- A coin has two sides – Head/Tail
 - 0/1
- Repeat tossing the coin 20 times
- Count the occurrences of Head and Tail, respectively.

Random Number Generator

□ rand()

- The rand function returns a pseudorandom integer in the range 0 to RAND_MAX (32767)

```
// Print 5 random numbers.  
for (int i = 0; i < 5; i++ )  
    cout << rand() << endl;
```

Seed of `rand()`

- With the same seed, the program will get the same result at each execution.
- Use `srand()` and choose the current time as the seed.

```
#include <time.h>
srand((unsigned) time(NULL));
for (int i = 0; i < 5; i++ )
    cout << rand() << endl;
```

Sample Code

(coin_tossing.cpp)

```
#include <iostream>
#include <time.h>
using std::cout;
using std::endl;
```

```
int main()
{
    int toss[2] = { 0 };           // 0 for Head; 1 for Tail
    int i;                         // coin tossing
    srand((unsigned) time(NULL));

    for (int k = 0; k < 20; k++ )
    {
        i = rand() % 2;
        toss[i] += 1;
    }

    cout << toss[0] << " Heads and " << toss[1] << " Tails.\n";
    return 0;
}
```

Structure of a Function

```
// Function to calculate x to the power n
double power(double x, int n)
{
    double result = 1.0;
    for (int i = 1; i<=n; i++)
        result *= x;

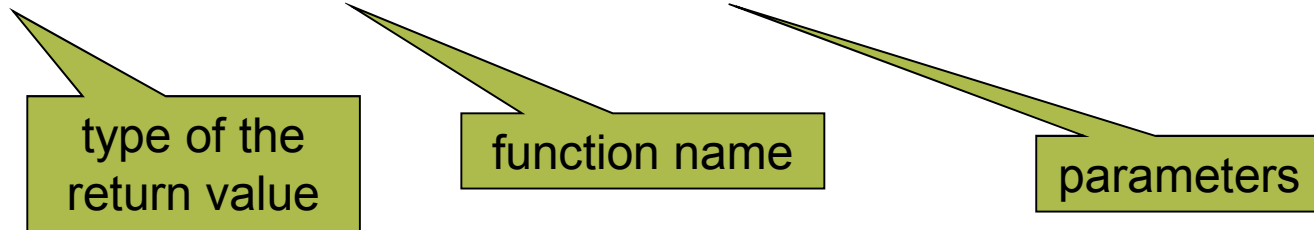
    return result;
}
```

} Function Header

} Function Body

The Function Header

```
double power ( double x, int n )
```



- ❑ The name of a function is governed by the same rules as those for a variable.
- ❑ A function returns either a single value, or nothing at all (`void`).
 - The single value returned can be a pointer, which contains the address of an array.

Arguments vs. Parameters

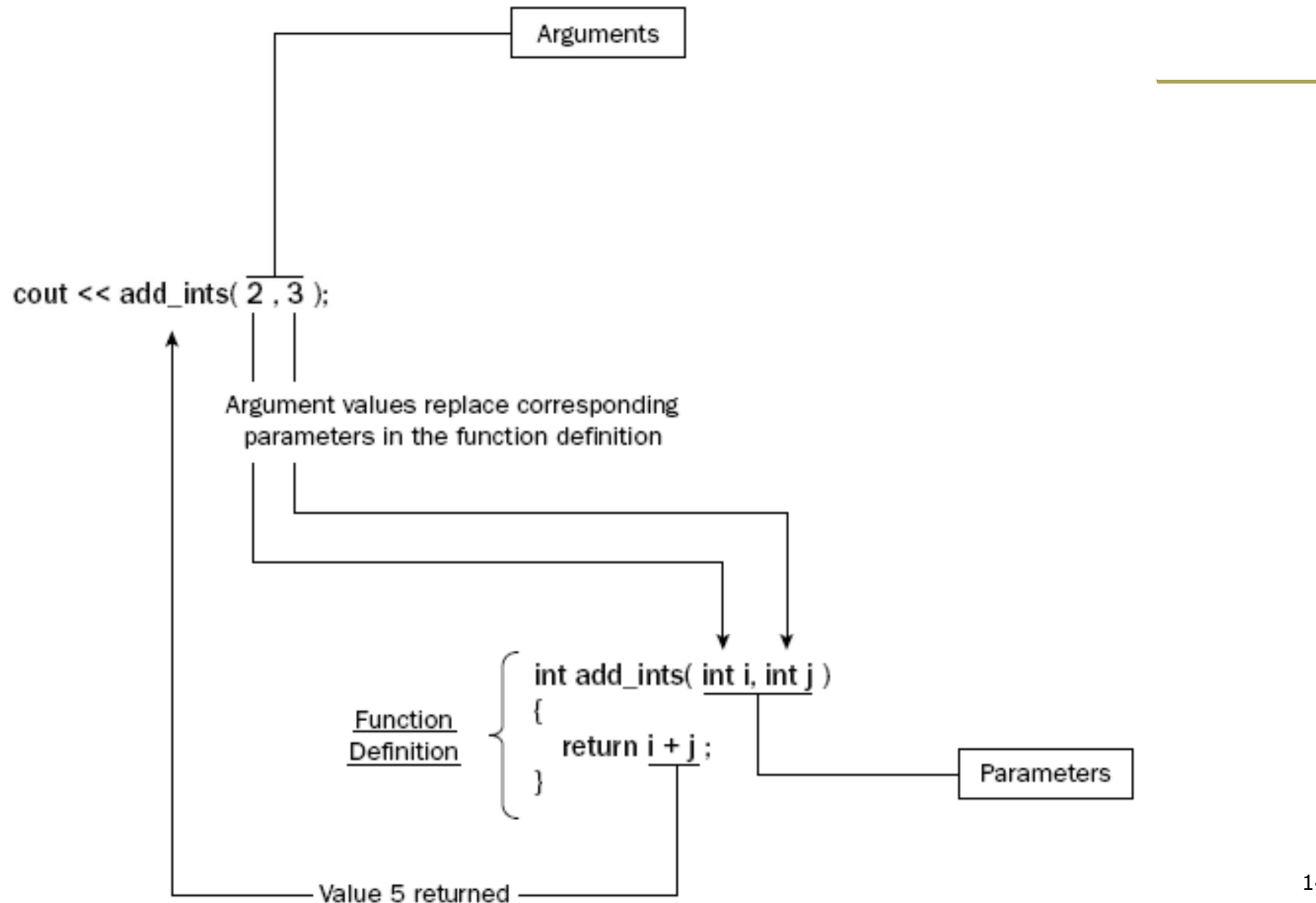


Figure 5-1

The Function Body

```
{
  double result = 1.0;
  for (int i = 1;
       i <= n; i++)
    result *= x;

  return result;
}
```

□ The `return` Statement

- Return a value (evaluated from an expression) to be the functional value.
- If the type of return value of this function is declared as `void`, there must be no expression.
 - `return;`

Using a Function

- ❑ Define the function before it is called.
- ❑ However, many programmers prefer to see `main()` earlier to have a global view.
 - Declare the function using a statement called a **function prototype**.

```
void print_stars()  
{  
    cout << "*****"  
        << endl;  
}  
  
int main()  
{  
    print_stars();  
    cout << "Test" << endl;  
    print_stars();  
}
```


C Math Functions

□ C numerics library

- `cmath` declares a set of functions to compute common mathematical operations and transformations
- `#include <cmath>` before you invoke the following functions.

Trigonometric functions

- cos - Compute cosine
- sin - Compute sine
- tan - Compute tangent
- acos - Compute arc cosine
- asin - Compute arc sine
- atan - Compute arc tangent
- atan2 - Compute arc tangent with two parameters

Hyperbolic functions

- cosh - Compute hyperbolic cosine
- sinh - Compute hyperbolic sine
- tanh - Compute hyperbolic tangent

Exponential and logarithmic functions:

- exp - Compute exponential function
- frexp - Get significand and exponent
- ldexp - Generate number from significand and exponent
- log - Compute natural logarithm
- log10 - Compute common logarithm
- modf - Break into fractional and integral parts

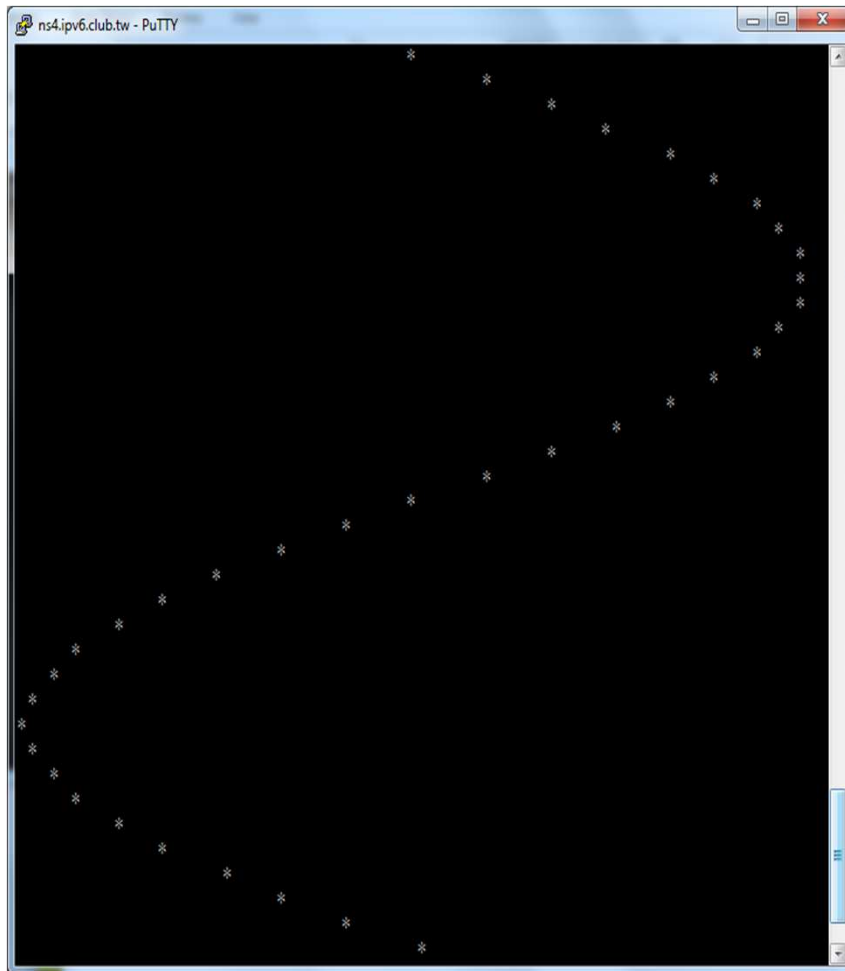
Power functions

- sqrt - Compute square root
- pow - Raise to power
 - `pow(64.0, 1.0/3.0)`

Rounding, absolute value and remainder functions:

- ceil - Round up value
- fabs - Compute absolute value
- floor - Round down value
- fmod - Compute remainder of division

Exercise: Draw a sine function



- Test your draw() function with the following main program.

```
int main()
{
    const float PI = 3.1415926535f;

    float x, y;
    for (int d=0; d<=360; d+=10)
    {
        x = d * PI/180;
        y = sin(x);
        draw(y);
    }
    return 0;
}
```