

## Predefined Functions

### Objectives of the Lecture

- Introduction
- Standard (predefined) functions
- Programming Example

### Introduction

- Functions are like **building blocks**.
- They allow complicated programs to be divided into **manageable pieces**.
- Some advantages of functions:
  - A programmer can focus on just that part of the program and construct it, debug it
  - Different people can work on different functions simultaneously
  - Can be re-used (even in different programs)
  - Enhance program readability
- Functions
  - Called **modules**
  - Can be put together to form a **larger program**

### Standard (Predefined Functions)

- In algebra, a function is defined as a rule or correspondence between values, called the function's **arguments**, and the unique value of the function associated with the arguments
  - If  $f(x) = 2x + 5$ :
    - then  $f(1) = 7$ ,  $f(2) = 9$ , and  $f(3) = 11$
    - 1, 2, and 3 are arguments
    - 7, 9, and 11 are the corresponding values
- Some of the predefined mathematical functions are:
  - `sqrt(x)`
  - `pow(x, y)`
  - `floor(x)`
- Predefined functions are organized into **separate libraries**
- I/O functions are in **iostream** header
- Math functions are in **cmath** header
- **pow(x,y)** calculates  $x^y$ 
  - `pow(2, 3) = 8.0`
  - Returns a value of type **double**
  - **x** and **y** are the parameters (or arguments)
    - The function has two parameters
- **sqrt(x)** calculates the nonnegative square root of **x**, for  $x \geq 0.0$ 
  - `sqrt(2.25) is 1.5`
  - Type **double**
- The floor function **floor(x)** calculates largest whole number not greater than x
  - `floor(48.79) is 48.0`
  - Type **double**
  - Has only one parameter

TABLE 6-1 Predefined Functions

Function	Header File	Purpose	Parameter(s) Type	Result
<code>abs(x)</code>	<code>&lt;cstdlib&gt;</code>	Returns the absolute value of its argument: <code>abs(-7) = 7</code>	<code>int</code>	<code>int</code>
<code>ceil(x)</code>	<code>&lt;cmath&gt;</code>	Returns the smallest whole number that is not less than <code>x</code> : <code>ceil(56.34) = 57.0</code>	<code>double</code>	<code>double</code>
<code>cos(x)</code>	<code>&lt;cmath&gt;</code>	Returns the cosine of angle <code>x</code> : <code>cos(0.0) = 1.0</code>	<code>double</code> (radians)	<code>double</code>
<code>exp(x)</code>	<code>&lt;cmath&gt;</code>	Returns $e^x$ , where $e = 2.718$ : <code>exp(1.0) = 2.71828</code>	<code>double</code>	<code>double</code>
<code>fabs(x)</code>	<code>&lt;cmath&gt;</code>	Returns the absolute value of its argument: <code>fabs(-5.67) = 5.67</code>	<code>double</code>	<code>double</code>

TABLE 6-1 Predefined Functions (continued)

Function	Header File	Purpose	Parameter(s) Type	Result
<code>floor(x)</code>	<code>&lt;cmath&gt;</code>	Returns the largest whole number that is not greater than <code>x</code> : <code>floor(45.67) = 45.00</code>	<code>double</code>	<code>double</code>
<code>islower(x)</code>	<code>&lt;cctype&gt;</code>	Returns <code>true</code> if <code>x</code> is a lowercase letter; otherwise, it returns <code>false</code> ; <code>islower('h')</code> is <code>true</code>	<code>int</code>	<code>int</code>
<code>isupper(x)</code>	<code>&lt;cctype&gt;</code>	Returns <code>true</code> if <code>x</code> is an uppercase letter; otherwise, it returns <code>false</code> ; <code>isupper('K')</code> is <code>true</code>	<code>int</code>	<code>int</code>
<code>pow(x, y)</code>	<code>&lt;cmath&gt;</code>	Returns $x^y$ ; if <code>x</code> is negative, <code>y</code> must be a whole number: <code>pow(0.16, 0.5) = 0.4</code>	<code>double</code>	<code>double</code>
<code>sqrt(x)</code>	<code>&lt;cmath&gt;</code>	Returns the nonnegative square root of <code>x</code> ; <code>x</code> must be nonnegative: <code>sqrt(4.0) = 2.0</code>	<code>double</code>	<code>double</code>
<code>tolower(x)</code>	<code>&lt;cctype&gt;</code>	Returns the lowercase value of <code>x</code> if <code>x</code> is uppercase; otherwise, it returns <code>x</code>	<code>int</code>	<code>int</code>
<code>toupper(x)</code>	<code>&lt;cctype&gt;</code>	Returns the uppercase value of <code>x</code> if <code>x</code> is lowercase; otherwise, it returns <code>x</code>	<code>int</code>	<code>int</code>

## Programming Example

```
// predefined functions.
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    int x;
    double u, v;
    u = 4.2;
    v = 3.0;
    cout << "\t " << u << " to the power of "
         << v << " = " << pow(u, v) << endl;
    cout << " 5.0 to the power of 4 = "
         << pow(5.0, 4) << endl;
    u = u + pow(3.0, 3);
    cout << " u = " << u << endl;
    x = -15;
    cout << ": Absolute value of " << x
         << " = " << abs(x) << endl;
    return 0;
}
```