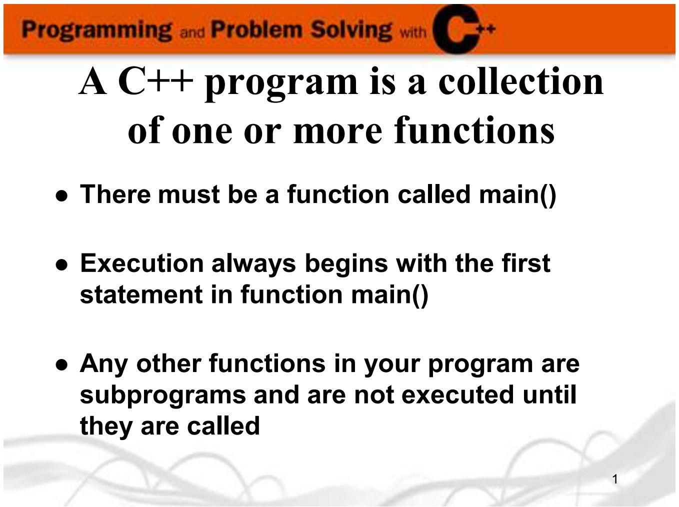
INTRODUCING C++ FUNCTIONS

VOID FUNCTIONS: **Never returns a value to its caller and is called as a separate statement**

VALUE RETURNING FUNCTION: **Always returns a single value to its caller and is called from within an expression (see sample.cpp)**

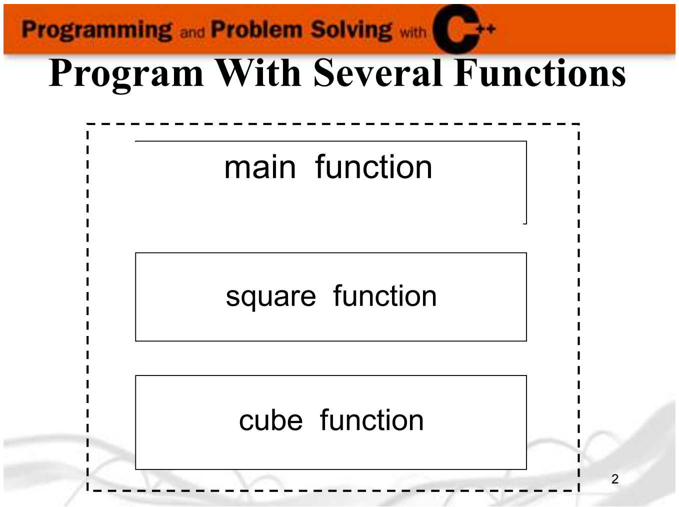
LIBRARY FUNCTIONS AND/OR USER OR PROGRAMMER DEFINED FUNCTIONS: UNDERSTAND HOW TO
(1) INCLUDE A LIBRARY FUNCTION AND/OR DECLARE A FUNCTION PROTOTYPE
(2) WRITE A FUNCTION DEFINITION
(3) MAKE A FUNCTION CALL



Programming and Problem Solving with C++

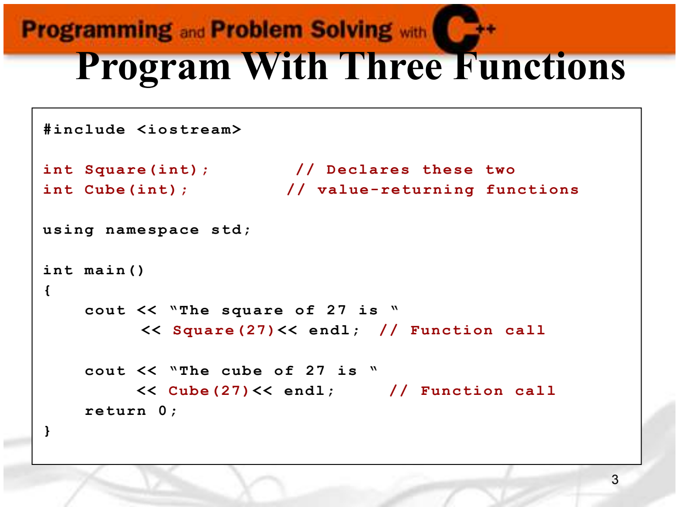
A C++ program is a collection of one or more functions

- **There must be a function called main()**
- **Execution always begins with the first statement in function main()**
- **Any other functions in your program are subprograms and are not executed until they are called**



Programming and Problem Solving with C++

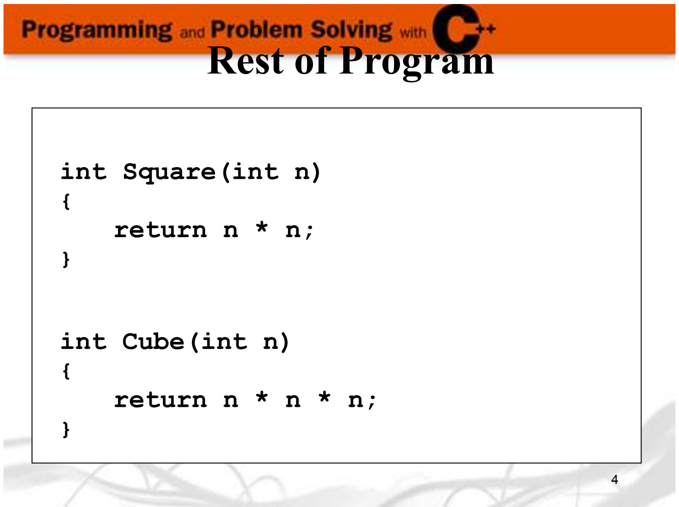
Program With Several Functions



Programming and Problem Solving with C++

Program With Three Functions

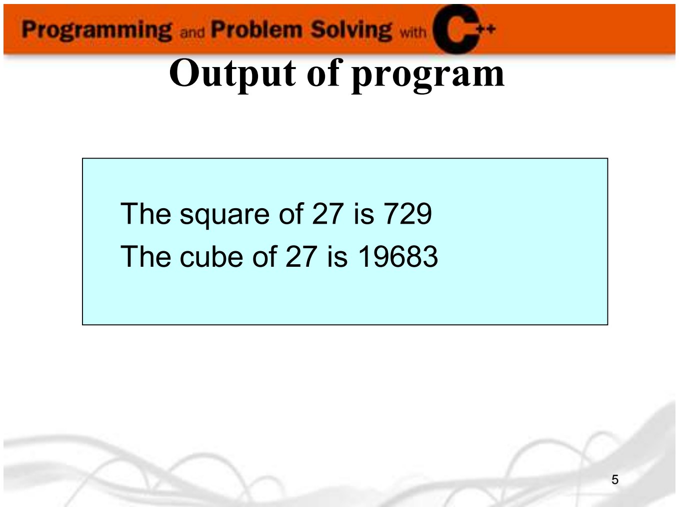
```
#include <iostream>

int Square(int);          // Declares these two
int Cube(int);           // value-returning functions

using namespace std;

int main()
{
     cout << "The square of 27 is "
          << Square(27)<< endl;  // Function call

     cout << "The cube of 27 is "
          << Cube(27)<< endl;      // Function call
     return 0;
}
```



Programming and Problem Solving with C++

Rest of Program

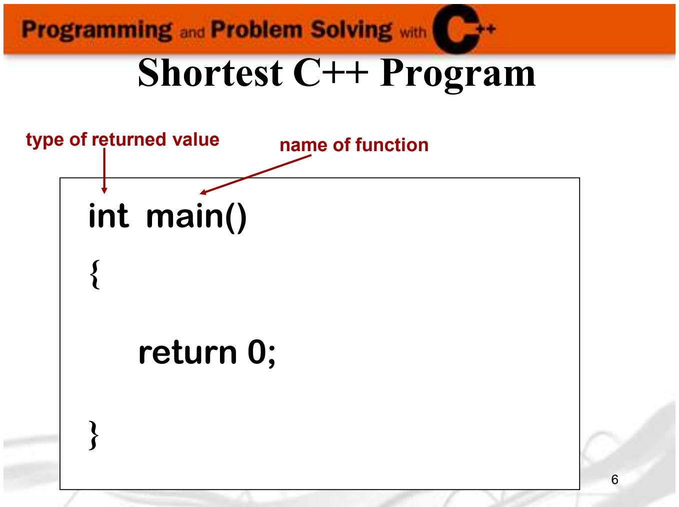
```
int Square(int n)
{
    return n * n;
}


int Cube(int n)
{
    return n * n * n;
}
```



Programming and Problem Solving with C++

Output of program

```
The square of 27 is 729
The cube of 27 is 19683
```



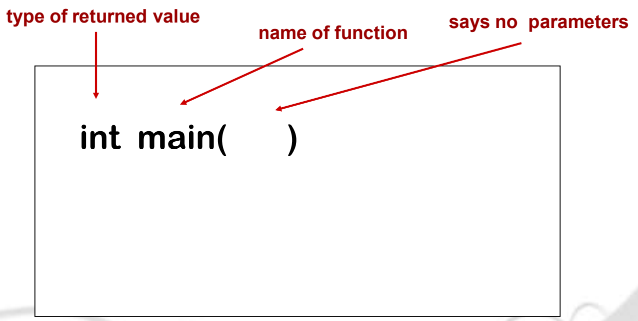
Programming and Problem Solving with C++

Shortest C++ Program

type of returned value
name of function

```
int main()
{

    return 0;

}
```

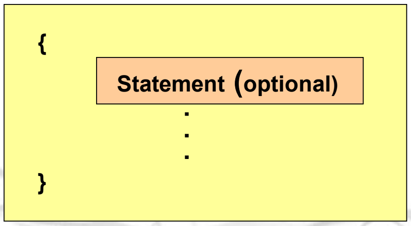
What is in a heading?



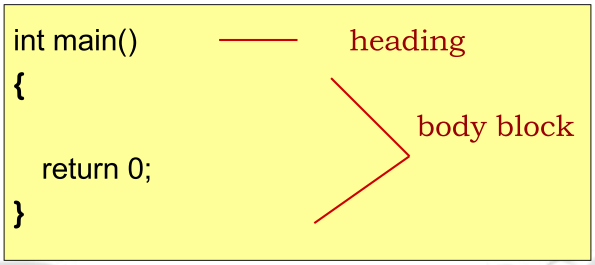
Block(Compound Statement)

- A **block** is a sequence of zero or more statements enclosed by a pair of curly braces { }

SYNTAX



Every C++ function has 2 parts



Functions

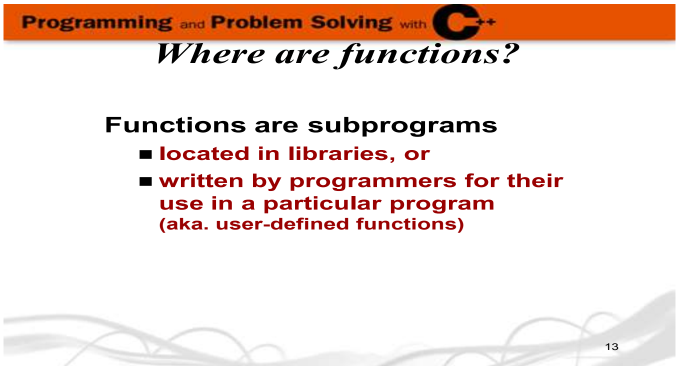
- Every C++ program must have a function called `main`
- Program execution always begins with function main
- Any other functions are subprograms and must be called

Function Calls

- One function calls another by using the name of the called function together with() containing an argument list
- A function call temporarily transfers control from the calling function to the called function

More About Functions

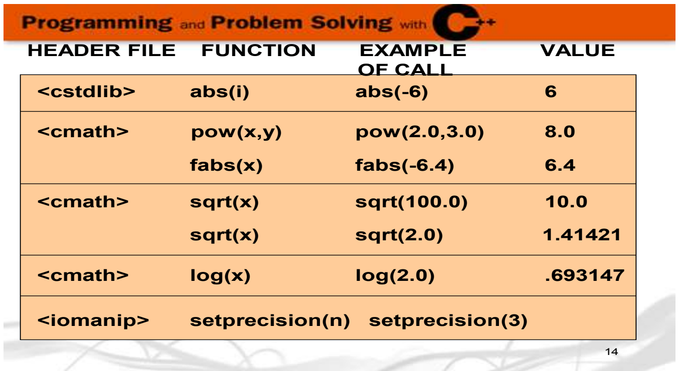
- It is not considered good practice for the body block of function main to be long
- Function calls are used to do subtasks
- Every C++ function has a return type
- If the return type is not void, the function returns a value to the calling block



Programming and Problem Solving with C++

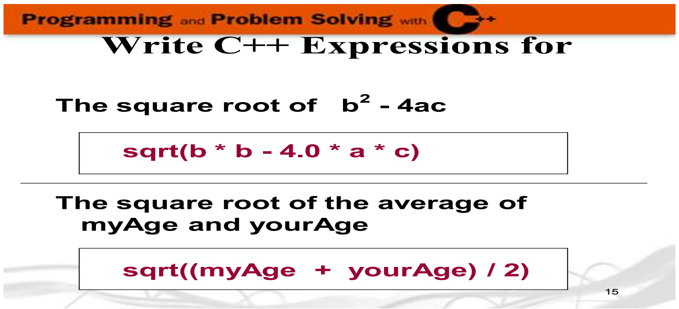
Where are functions?

Functions are subprograms
- located in libraries, or
- written by programmers for their use in a particular program (aka. user-defined functions)



Programming and Problem Solving with C++

HEADER FILE	FUNCTION	EXAMPLE OF CALL	VALUE
<cstdlib>	abs(i)	abs(-6)	6
<cmath>	pow(x,y)	pow(2.0,3.0)	8.0
	fabs(x)	fabs(-6.4)	6.4
<cmath>	sqrt(x)	sqrt(100.0)	10.0
	sqrt(x)	sqrt(2.0)	1.41421
<cmath>	log(x)	log(2.0)	.693147
<iomanip>	setprecision(n)	setprecision(3)	



Programming and Problem Solving with C++

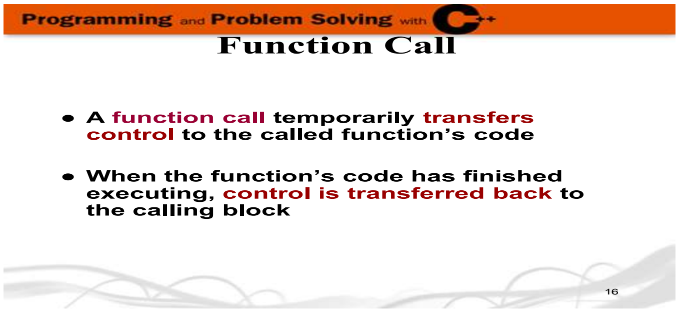
Write C++ Expressions for

The square root of $b^2 - 4ac$

sqrt(b * b - 4.0 * a * c)

The square root of the average of myAge and yourAge

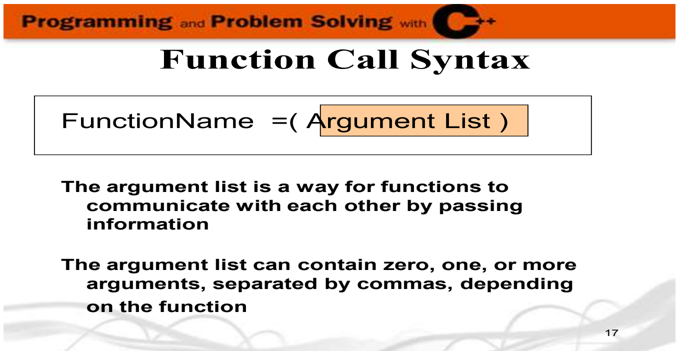
sqrt((myAge + yourAge) / 2)



Programming and Problem Solving with C++

Function Call

- A function call temporarily transfers control to the called function's code
- When the function's code has finished executing, control is transferred back to the calling block



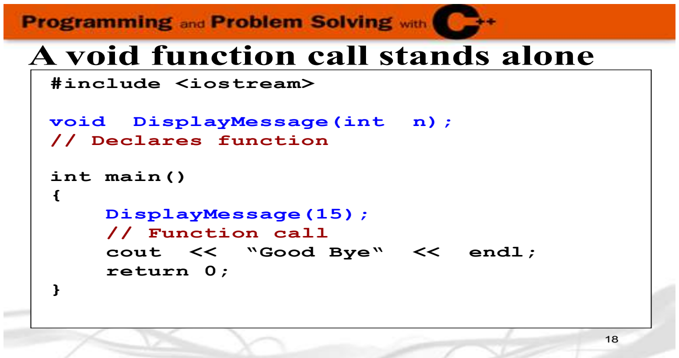
Programming and Problem Solving with C++

Function Call Syntax

FunctionName =(Argument List)

The argument list is a way for functions to communicate with each other by passing information

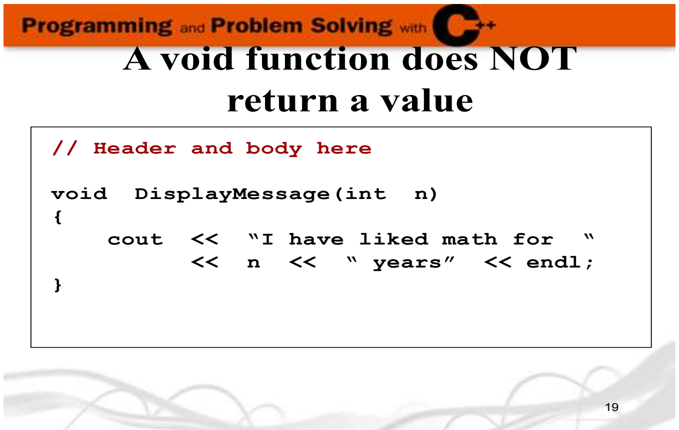
The argument list can contain zero, one, or more arguments, separated by commas, depending on the function



Programming and Problem Solving with C++

A void function call stands alone

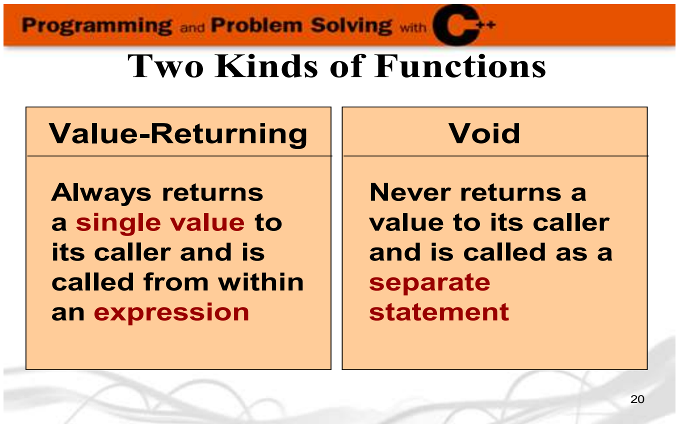
```
#include <iostream>

void  DisplayMessage(int  n);
// Declares function

int main()
{
    DisplayMessage(15);
    // Function call
    cout  <<  “Good Bye“  <<  endl;
    return 0;
}
```



Programming and Problem Solving with C++

A void function does NOT return a value

```
// Header and body here

void  DisplayMessage(int  n)
{
    cout  <<  “I have liked math for  ”
          <<  n  <<  “ years”  << endl;
}
```



Programming and Problem Solving with C++

Two Kinds of Functions

Value-Returning	Void
Always returns a single value to its caller and is called from within an expression	Never returns a value to its caller and is called as a separate statement

Files used to demonstrate C++ function types topic using a C++ compiler

Sample.cpp - demonstrates the set-up and calls for two user-defined functions

ch3math1.cpp - demonstrates the set-up and call of a pre-programmed math function from the standard library

ch3math1a.cpp - demonstrates the set-up and call of a pre-programmed math function from the standard library as well as the set-up and call for a user-defined void function

MILES_a.cpp – demonstrates the set-up and call of a user-defined value-returning

MILES_b.cpp – demonstrates the set-up and call of both user-defined value-returning as well as a void function. Pay particular attention to how both the functions are set-up and how they are called.

```
//*****************************************
// Sample program –sample.cpp
// This program computes the square and cube of 27
//*****************************************
#include <iostream>
using namespace std;
int Square( int );
int Cube( int );
int main()
{
 cout << "The square of 27 is " << Square(27) << endl;
   cout << "and the cube of 27 is " << Cube(27) <<
endl;

  system("PAUSE");
   return 0;
}

int Square( int n )
{
   return n * n;
}

int Cube( int n )
{
   return n * n * n;
}
```

```
//*****************************************************************
// chapter 3 - Value-Returning Functions and Void Functions – ch3math1a.cpp
//the value-returning function is accessed from cmath ..a stand library file
//while the void function is a user-defined function
//take note of the set-up and the call statements for these two functions
//*****************************************************************
#include <cmath>                    //for math functions
#include <iostream>

using namespace std;

void Hdr ();

int main()
{


   Hdr();   //void function call

   cout <<"pow (3.0, 4.0) =" << pow (3.0, 4.0)<<endl;
   cout <<"sqrt (81.0) =" << sqrt (81.0) <<endl;

   system("PAUSE");
   return 0;
}

//********************************
void Hdr ()
{
cout <<"//************************************************"<<endl;
cout << "// chapter 3 - Value-Returning Functions Example#1" <<endl;
cout << "// Using preprogrammed functions from the standard library" <<endl;
cout << "//***********************************************" <<endl;
}
```

Practice writing functions WEEK#4….

(1) Pseudo code (design phase or algorithm development) to writing, testing and debugging source code (implementation phase)

Convert the following pseudo code to C++ code. Be sure to define the appropriate variables.

Set-up variables to store three test scores.

Provide appropriate **input prompts and get three values** from the keyboard.

Set-up a value-returning function called *float average (int,int,int)* that will return the average of three values. This step should include the **function prototype statement** and the **function definition**.

Write the **function call** statement in main to call *average (score1, score2, score3)* …making sure that you send in the three test score to the function.

Show or **display the average** of the three scores

Note: All the predefined functions require using namespace std; as well as the appropriate include directives

*Use #include<cmath> for sqrt, pow, fabs, ceil, floor and #include<cstdlib> for abs, labs, exit, rand, srand

(2) Using appropriate #include directives write code snippets to determine the value of each of the following arithmetic expressions.

sqrt(16.0)   sqrt(16)   pow(2.0, 3.0)   pow(2, 3)   pow(2.0, 3)   pow(1.1, 2)

abs(3)   abs(-3)   abs(0)   fabs(-3.0)   fabs(-3.5)   fabs(3.5)

ceil(5.1)   ceil(5.8)   floor(5.1)   floor(5.8)   pow(3.0,2)/2.0   pow(3.0,2)/2

7/abs(-2)   (7+ sqrt(4.0))/3.0   sqrt(pow(3, 2))

(3) Write separate user-defined value-returning functions that contain the following mathematical expressions written as C++ arithmetic expressions. Each function call should provide the appropriate arguments and return the computed value to be displayed on the screen.

a. $\sqrt{x+y}$ b. x^{y+2} c. $\sqrt{area + fudge}$

d. $\frac{\sqrt{time - tide}}{nobody}$ e. $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ f. $\frac{\pi}{2}$

Programming Challenges to Review with updates to include functions (Lecture/Lab – week# 4)

(1) Test Average

Write a program that asks for five test scores. The program should calculate the average test score and display it. The number displayed should be formatted in fixed-point notation, with one decimal point of precision.

Update your program to include a value returning function that takes as arguments the five test scores and returns the average test score.

(2A) Average Rainfall

Write a program that calculates the average monthly rainfall for three months. The program should ask the user to enter the name of each month, such as June or July, and the amount of rain (in inches) that fell that month. The program should display a message similar to the following:

The average monthly rainfall for June, July, and August was 6.72 inches.

Update your program to include a void function to get the necessary user inputs of the names of the months and the rainfall amount for the month.

(2B) Using Files—Average Rainfall Modification

Modify the program you wrote for Programming Challenge (Average Rainfall) so the void functions now reads its input from a file instead of from the keyboard. Sample data to test your program can be found in the rainfall.dat file.

Use notepad or the text editor in your compiler to create a new data file and insert the data below and save the file as rainfall.dat

```
April   .87
May     1.51
June    1.05
```

(3) Using Files—Storing and Retrieving Numbers

For this assignment you will write two programs:

Program 1: Write a program that asks the user to enter five floating-point numbers.
Then write a void function that will create a file and save all five numbers to the file.

Program 2: Write a program that opens the file created by Program 1, reads the five numbers, and displays them. The program should also include a value returning function that will calculate and display the sum of the five numbers.

NOTES: