PEN203

C++ Functions

C++ How to Program
Deitel & Deitel

Outline

- Program Modules in C++
- Math Library Functions
- Functions
- Function Definitions
- Function Prototypes
- Calling Functions: Call by Value and Call by Reference
- Random Number Generation
- Recursion

Program Modules in C++

- C++ programs can call user-defined functions and built in library functions.
- A function is called by function name and argument
- Function performs operations and returns results
- Functions can be considered as modules in C++

Math Library Functions

- Used to perform math computations
- To be able to use math library functions, C++ proprams should include <cmath> (#include <cmath>)
- Example:

```
cout<<pow( 5, 2 );
```

- All math functions return double data
- Arguments may be constants, variables, or expressions.

Functions

- Functions inherently modularize programs
- The variables defined in function definition are called local variables and they are only be accessed in function.
- Function parameters are also local variables. They are used to communicate between functions and they are also local variables.

Functions

- Advantages of Functions
  - Manageable program development
  - Software reusability
  - Avoid code repetition

Function Definitions

- Function definition format

```
return-value-type function-name(parameter-list)
{
        declarations and statements
}
```

- void as a return type indicates that function returns nothing
- Parameters given as a comma seperated list.
- Functions can not be defined inside other functions.
- If the function returns nothing, only return; or nothing is provided.

Function Definitions

```cpp
// Fig. 3.3: fig03_03.cpp
// Creating and using a programmer-defined function.
#include <iostream>

using std::cout;
using std::endl;

int square( int );   // function prototype

int main()
{
   // loop 10 times and calculate and output
   // square of x each time
   for ( int x = 1; x <= 10; x++ )
      cout << square( x ) << "  ";  // function call

   cout << endl;

   return 0;  // indicates successful termination

} // end main

// square function definition returns square of an integer
int square( int y )  // y is a copy of argument to function
{
   return y * y;     // returns square of y as an int

} // end function square
```

Function Prototypes

- Function prototype includes:
  - Function name
  - Parameters
  - Return type
- Prototypes are needed if the function definition is provided after main program.
- Example:

  int maximum(int x, int y, int z),

  The maximum function takes 3 integers and returns integer value as a result.

Calling Functions: Call by Value and Call by Reference

- Call by value
  - A copy of the argument is created and passed to function.
  - Modifications performed in function do not effect the original value.
- Call by reference
  - Original argument passed to function
  - Modifications in function effect the original value.