Programmer-Defined Functions

Lecture 16
Sections 6.1 - 6.4

Robb T. Koether

Hampden-Sydney College

Wed, Oct 2, 2019

Outline

Programmer-Defined Functions

- A programmer may create his own functions.
- If his function is invoked in `main()`, then
  - Execution leaves `main()` (which is itself a function) and goes to beginning of the function.
  - The function executes its statements from top to bottom or until it hits a **`return`** statement, just as in `main()`.
  - When the function is finished, execution returns to its departure point in `main()` and continues.

Outline

The Form of a Function

The Form of a Function

```
return-type function-name(formal-param-list)
{
    function-body
    return return-value;
}
```

- function-name is the name that the programmer chooses for the function.
- The *formal-param-list* is a list of object types and names (i.e., declarations) passed to the function as parameters.
- The function-body is a sequence of C++ statements that will compute the appropriate value for the returned object.

The Form of a Function

The Form of a Function

```
return-type function-name(formal-param-list)
{
    function-body
    return return-value;
}
```

- return-type is the type of object that the function returns to the calling function.
- return-value must be of type *return-type*.

The Form of a Function

The Form of a Function

```
void function-name(formal-param-list)
{
    function-body
    return          // Optional - no value given
}
```

- The return type may be **`void`**, in which case the **`return`** statement is optional.
- If it is written, then there is no return value specified.

Outline

Example of a Function

Function Definition

```
float average3(float a, float b, float c)
{
    float avg = (a + b + c)/3.0f;
    return avg;
}
```

- This function will return the average of three numbers.

Example of a Function

Function Definition

```
float average3(float a, float b, float c)
{
    return (a + b + c)/3.0f;
}
```

- It is legal to return the value of an expression.

Example of a Function

Function Definition

```
bool isOdd(int n)
{
    return n % 2 == 1;
}
```

- This function will return the average of three numbers.

Outline

Function Calls

Function Call

```
function-name(actual-param-list)
```

- To call a function, we
  - Name the function, and
  - Give the *actual-param-list* (a list of actual objects or expressions).
- A function call occurs in an expression.
- If the function is **`void`** type, then the function call is a complete statement and must occur on a line by itself.
- If the function is not **`void`** type, then the function call occurs within an expression where an object of the function's return type is permitted.

Function Calls

- If the function is **`void`** type, then the function call is a complete statement and must occur on a line by itself.
- If the function is not **`void`** type, then the function call occurs within an expression where an object of the function's return type is permitted.

Function Usage

- Data types are not specified in the actual parameter list.
- Actual vs. formal parameters
  - Types must match, or else
  - There must be a conversion rule to convert the actual type into the formal type, the same as with assignment statements.

Example of Function Usage

Function Usage

```
cout << "Enter the three grades:  ";
float grade1, grade2, grade3;
cin >> grade1 >> grade2 >> grade3;
float avg_grade = average3(grade1, grade2, grade3)
cout << avg_grade << endl;
```

- This program finds the average of three grades.

Example of Function Usage

Function Usage

```
cout << "Enter the three grades:  ";
float grade1, grade2, grade3;
cin >> grade1 >> grade2 >> grade3;
cout << average3(grade1, grade2, grade3) << endl;
```

- The function call may be placed in the output statement, although this can make the code harder to read.

Outline

Prototypes

- Before the compiler can check and compile a function call, it must have already seen the function's prototype.
- Typically, we place the prototypes of all of the functions before `main()`.
- Then we are free to use them anywhere in the program.

Example of a Prototype

Function Prototype

```
float average3(float a, float b, float c);

int main()
{
⋮

float avg = average3(x, y, z)
⋮

}

float average3(float a, float b, float c)
{
return (a + b + c)/3.0f;
}
```

- The prototype of the `average()` function.

Outline

Examples

- Create and run
  - `Average3.cpp`
  - `Hypotenuse.cpp`
  - `SquareRoot.cpp`

Outline

Example of Header File

- Example
  - `isVowelFunc.cpp`, `isvowel.cpp`, `isvowel.h`

Outline

Assignment

Assignment

- Read Sections 6.1 - 6.4.