

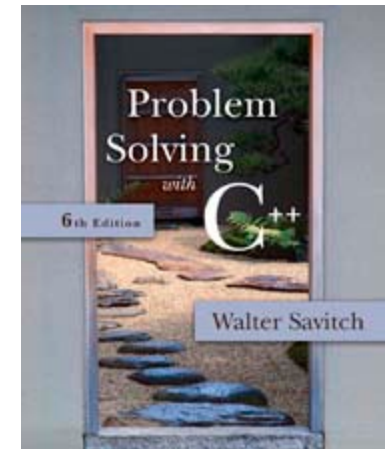
APS105: Lecture 9

Wael Aboelsaadat

wael@cs.toronto.edu

<http://ccnet3.utoronto.ca/20079/aps105h1f/>

Acknowledgement: These slides are a modified version of the text book slides as supplied by Addison Wesley



Constants

- `const` is the keyword to declare a constant

- Example:

```
const int WINDOW_COUNT = 10;
```

declares a constant named `WINDOW_COUNT`

- Its value cannot be changed by the program like a variable
- It is common to name constants with all capitals

Display 2.16

Comments and Named Constants

```
//File Name: health.cpp (Your system may require some suffix other than cpp.)  
//Author: Your Name Goes Here.  
//Email Address: you@yourmachine.bla.bla  
//Assignment Number: 2  
//Description: Program to determine if the user is ill.  
//Last Changed: September 23, 2004
```

```
#include <iostream>  
using namespace std;  
int main()  
{  
    const double NORMAL = 98.6;//degrees Fahrenheit  
    double temperature;  
  
    cout << "Enter your temperature: ";  
    cin >> temperature;  
  
    if (temperature > NORMAL)  
    {  
        cout << "You have a fever.\n";  
        cout << "Drink lots of liquids and get to bed.\n";  
    }  
    else  
    {  
        cout << "You don't have a fever.\n";  
        cout << "Go study.\n";  
    }  
  
    return 0;  
}
```

*Your programs should always
begin with a comment
similar to this one.*

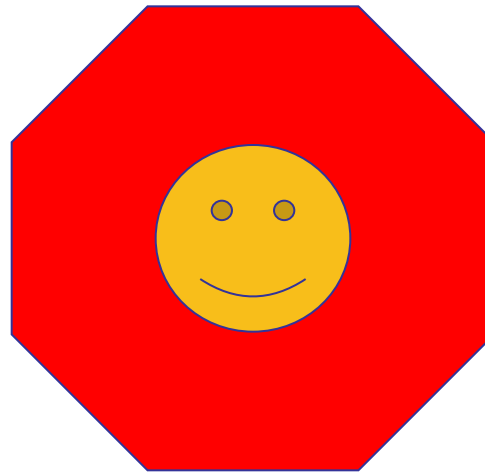
Display 2.16



Sample Dialogue

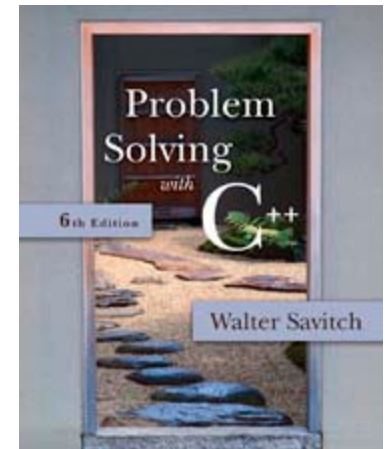
Enter your temperature: **98.6**
You don't have a fever.
Go study.

Chapter 2 -- End



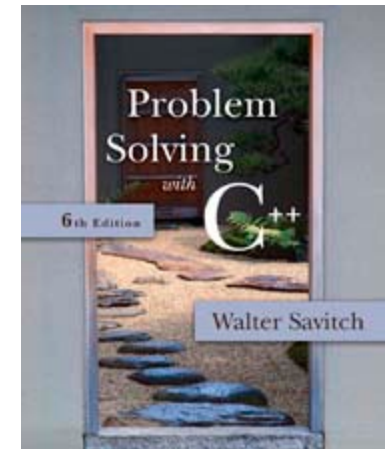
Chapter 4

Procedural Abstraction and Functions That Return a Value



4.2

Predefined Functions



Functions in math

$$f(x) = \sin(x+10) + \cos(\tan(x)) + \log(x)$$

what is $f(60)$?

$$f(60) = 0.94 + 0.99 + 1.77 = 3.7$$

How to write a function?

function-name(parameter1, parameter2,)

Result = function-name(parameter1, parameter2,)

Predefined Functions

- C++ comes with libraries of predefined functions
- Example: `sqrt` function
 - `the_root = sqrt(9.0);`
 - returns, or computes, the square root of a number
 - The number, 9, is called the argument
 - `the_root` will contain 3.0

Function Calls

- `sqrt(9.0)` is a function call
 - It invokes, or sets in action, the `sqrt` function
 - The argument (9), can also be a variable or an expression
- A function call can be used like any expression
 - `bonus = sqrt(sales) / 10;`
 - `Cout << "The side of a square with area " << area`
`<< " is "`
`<< sqrt(area);`

Display 4.1

Function Call Syntax

- `Function_name (Argument_List)`
 - `Argument_List` is a comma separated list:

`(Argument_1, Argument_2, ... ,
Argument_Last)`

- Example:
 - `side = sqrt(area);`
 - `cout << "2.5 to the power 3.0 is "
 << pow(2.5, 3.0);`

Function Libraries

- Predefined functions are found in libraries
- The library must be “included” in a program to make the functions available
- An include directive tells the compiler which library header file to include.
- To include the math library containing `sqrt()`:

```
#include <cmath>
```

- Newer standard libraries, such as `cmath`, also require the directive

```
using namespace std;
```

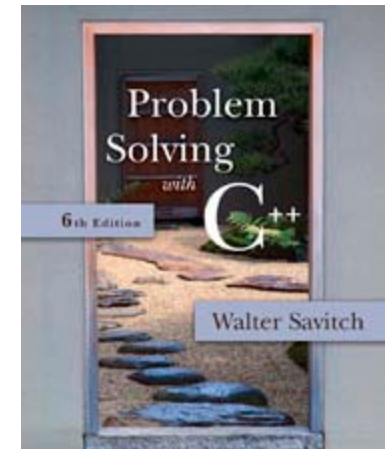
Other Predefined Functions

- `abs(x)` --- `int value = abs(-8);`
 - Returns absolute value of argument `x`
 - Return value is of type `int`
 - Argument is of type `x`
 - Found in the library `cstdlib`
- `fabs(x)` --- `double value = fabs(-8.0);`
 - Returns the absolute value of argument `x`
 - Return value is of type `double`
 - Argument is of type `double`
 - Found in the library `cmath`

Display 4.2

4.3

Programmer-Defined Functions



Programmer-Defined Functions

- Two components of a function definition
 - Function declaration (or function prototype)
 - Shows how the function is called
 - Must appear in the code before the function can be called
 - Syntax:
`Type_returned Function_Name(Parameter_List);`
`//Comment describing what function does`

- Function definition
 - Describes how the function does its task
 - Can appear before or after the function is called
 - Syntax:
`Type_returned Function_Name(Parameter_List)`
`{`
`//code to make the function work`
`}`

Function Definition

- Provides the same information as the declaration
- Describes how the function does its task

function header

- Example:

```
double total_cost(int number_par, double price_par)
{
    const double TAX_RATE = 0.05; //5% tax
    double subtotal;
    subtotal = price_par * number_par;
    return (subtotal + subtotal * TAX_RATE);
}
```

function body

The Return Statement

- Ends the function call
- Returns the value calculated by the function
- Syntax:

return expression;

- expression performs the calculation
or
- expression is a variable containing the
calculated value

- Example:

return subtotal + subtotal * TAX_RATE;

The Function Call

- Tells the name of the function to use
- Lists the arguments
- Is used in a statement where the returned value makes sense
- Example:

```
double bill = total_cost(number, price);
```

Display 4.3

Can you rewrite this program using functions?

```
#include <iostream>

using namespace std;

int main( )
{
    char ans      = ' ';
    bool bContinue = true;
    bool bError   = false;

    do
    {
        if( bError != true )
            cout << "\nHello\n";

        cout << "\nDo you want another greeting?\n"
              << "Press y for yes, n for no, \n"
              << "and then press return: ";

        cin >> ans;

        if( ans=='n' || ans == 'N')
        {
            bError   = false;
            bContinue = false;
        }
        else
        {
            if( ans=='y' || ans == 'Y')
            {
                bError   = false;
                bContinue = true;
            }
            else
            {
                bError   = true;
                bContinue = true;
                cout << "\nYou have entered an invalid input\n";
            }
        }
    } while( bContinue );

    cout << "\nGood-Bye\n";
    return 0;
}
```

A Function Call

```
//Computes the size of a dog house that can be purchased
//given the user's budget.
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    const double COST_PER_SQ_FT = 10.50;
    double budget, area, length_side;

    cout << "Enter the amount budgeted for your dog house $";
    cin >> budget;

    area = budget/COST_PER_SQ_FT;
    length_side = sqrt(area);

    cout.setf(ios::fixed);
    cout.setf(ios::showpoint);
    cout.precision(2);
    cout << "For a price of $" << budget << endl
         << "I can build you a luxurious square dog house\n"
         << "that is " << length_side
         << " feet on each side.\n";

    return 0;
}
```

Sample Dialogue

```
Enter the amount budgeted for your dog house $25.00
For a price of $25.00
I can build you a luxurious square dog house
that is 1.54 feet on each side.
```

Display 4.1



Display 4.2



Some Predefined Functions

Name	Description	Type of Arguments	Type of Value Returned	Example	Value	Library Header
sqrt	square root	<i>double</i>	<i>double</i>	sqrt(4.0)	2.0	cmath
pow	powers	<i>double</i>	<i>double</i>	pow(2.0,3.0)	8.0	cmath
abs	absolute value for <i>int</i>	<i>int</i>	<i>int</i>	abs(-7) abs(7)	7 7	cstdlib
labs	absolute value for <i>long</i>	<i>long</i>	<i>long</i>	labs(-70000) labs(70000)	70000 70000	cstdlib
fabs	absolute value for <i>double</i>	<i>double</i>	<i>double</i>	fabs(-7.5) fabs(7.5)	7.5 7.5	cmath
ceil	ceiling (round up)	<i>double</i>	<i>double</i>	ceil(3.2) ceil(3.9)	4.0 4.0	cmath
floor	floor (round down)	<i>double</i>	<i>double</i>	floor(3.2) floor(3.9)	3.0 3.0	cmath

A Function Definition (part 1 of 2)

```
#include <iostream>
using namespace std;
```

```
double total_cost(int number_par, double price_par); ← function declaration
//Computes the total cost, including 5% sales tax,
//on number_par items at a cost of price_par each.
```

```
int main()
{
    double price, bill;
    int number;

    cout << "Enter the number of items purchased: ";
    cin >> number;
    cout << "Enter the price per item $";
    cin >> price;
    bill = total_cost(number, price); ← function call

    cout.setf(ios::fixed);
    cout.setf(ios::showpoint);
    cout.precision(2);
    cout << number << " items at "
         << "$" << price << " each.\n"
         << "Final bill, including tax, is $" << bill
         << endl;

    return 0;
}
```

```
double total_cost(int number_par, double price_par) ← function heading
{
    const double TAX_RATE = 0.05; //5% sales tax
    double subtotal;

    subtotal = price_par * number_par;
    return (subtotal + subtotal*TAX_RATE);
} ← function body → function definition
```

Display 4.3 (1/2)

