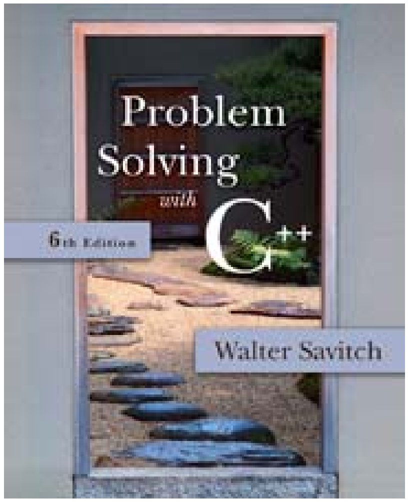
APS105: Lecture 8

Wael Aboelsaadat

wael@cs.toronto.edu

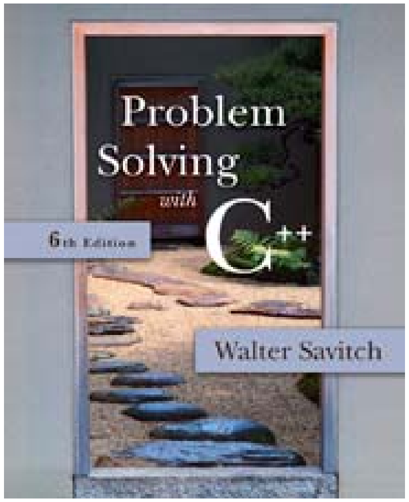
http://ccnet3.utoronto.ca/20079/aps105h1f/

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



2.4

Simple Flow of Control



while Loop Syntax

- while (boolean expression is true)

  ```
  {
          statement1;
          statement2;
          ….
  }
  ```

  - Semi-colons are used only to end the statements within the loop
- While (boolean expression is true)

  ```
          statement to repeat
  ```

Display 2.11

do-while loop

- A variation of the while loop.
- A do-while loop is always executed at least once
  - The body of the loop is first executed
  - The boolean expression is checked after the body has been executed
- Syntax:

```
do
{
    statement1;
    statement2;
    ….
}  while (boolean_expression);
```

Display 2.12

Display 2.13

Example: reading Y/N and detecting error

```
#include <iostream>

using namespace std;

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

   do
   {
           if( bError == false )
               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 == true );

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

Infinite Loops

- Loops that never stop are infinite loops
- The loop body should contain a line that will eventually cause the boolean expression to become false
- Example: Print the odd numbers less than 12

```
x = 1;
while (x != 12)
      {
          cout << x << endl;
          x = x + 2;
       }
```

- Better to use this comparison: while (x < 12)

Section 2.4 Conclusion

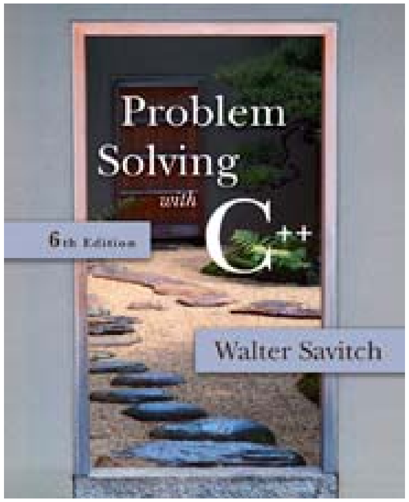
- Can you
  - Tell the output of this code if x is of type int?

```
x = 10;
while ( x > 0)
  {
        cout << x << endl;
        x = x – 3;
    }
```

  - Tell the output of the previous code using the comparison x < 0 instead of x > 0?

2.5

Program Style



Program Style

- A program written with attention to style
  - is easier to read
  - easier to correct
  - easier to change

Program Style - Indenting

- Items considered a group should look like a group
  - Skip lines between logical groups of statements
  - Indent statements within statements

```
if (x = = 0)
    statement;
```

- Braces {} create groups
  - Indent within braces to make the group clear
  - Braces placed on separate lines are easier to locate

Program Style – Indenting – which is better?

```
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;
}
```

```
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 inp
                    }
```

Program Style - Comments

- // is the symbol for a single line comment
  - Comments are explanatory notes for the programmer
  - All text on the line following // is ignored by the compiler
  - Example:

    ```
    //calculate regular wages
    gross_pay = rate * hours;
    ```
- /* and */ enclose multiple line comments
  - Example:

    ```
    /*  This is a comment that spans
         multiple lines without a
         comment symbol on the middle line
    */
    ```

Program Style - Comments

- // is the symbol for a single line comment
  - Comments are explanatory notes for the programmer
  - All text on the line following // is ignored by the compiler
  - Example:
    ```
    //calculate regular wages
    gross_pay = rate * hours;
    ```
- /* and */ enclose multiple line comments
  - Example:
    ```
    /*  This is a comment that spans
          multiple lines without a
          comment symbol on the middle line
    */
    ```

Program Style – Variable names

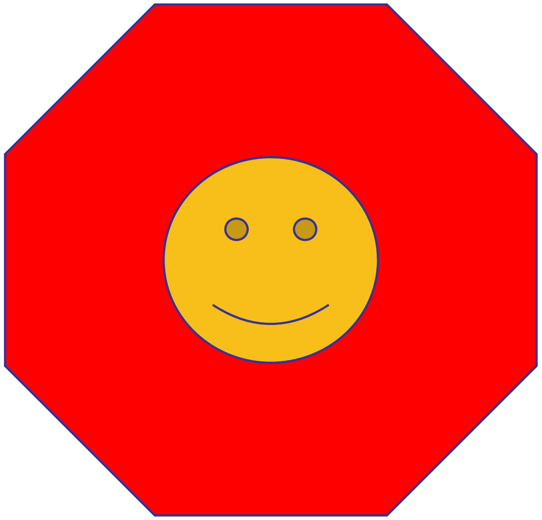
- Use meaningful names

```
int A,B,C,MyVariable;
      vs.
int Height,Weight,Density,Volume;
```

- Use Hungarian notation

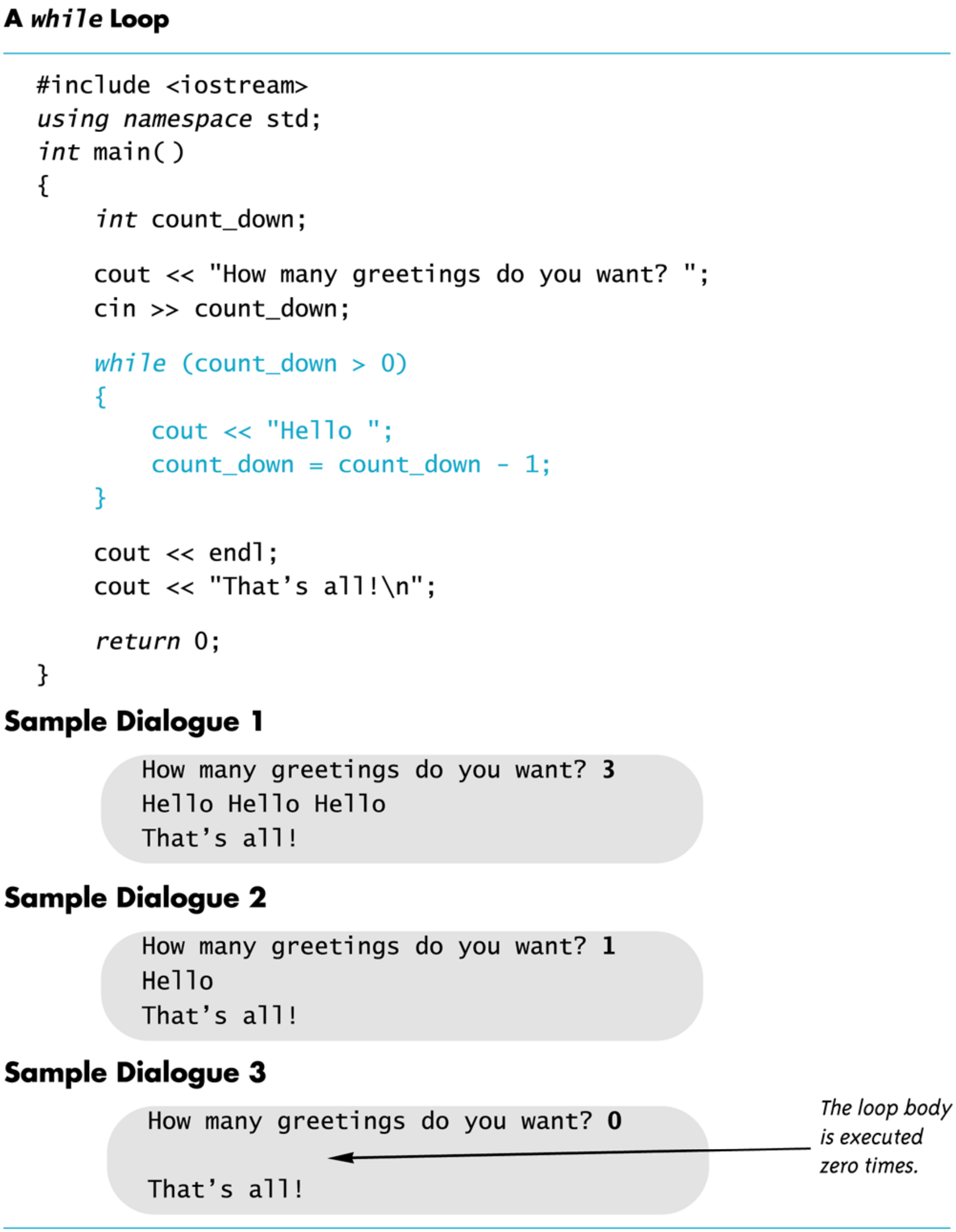
```
int Height,Weight,Density,Volume;
      vs.
int iHeight,iWeight,iDensity,iVolume;
```

Chapter 2 -- End



Display 2.11

Back Next



A *while* Loop

```
#include <iostream>
using namespace std;
int main( )
{
    int count_down;

    cout << "How many greetings do you want? ";
    cin >> count_down;

    while (count_down > 0)
    {
        cout << "Hello ";
        count_down = count_down - 1;
    }

    cout << endl;
    cout << "That's all!\n";

    return 0;
}
```

Sample Dialogue 1

```
How many greetings do you want? 3
Hello Hello Hello
That's all!
```

Sample Dialogue 2

```
How many greetings do you want? 1
Hello
That's all!
```

Sample Dialogue 3

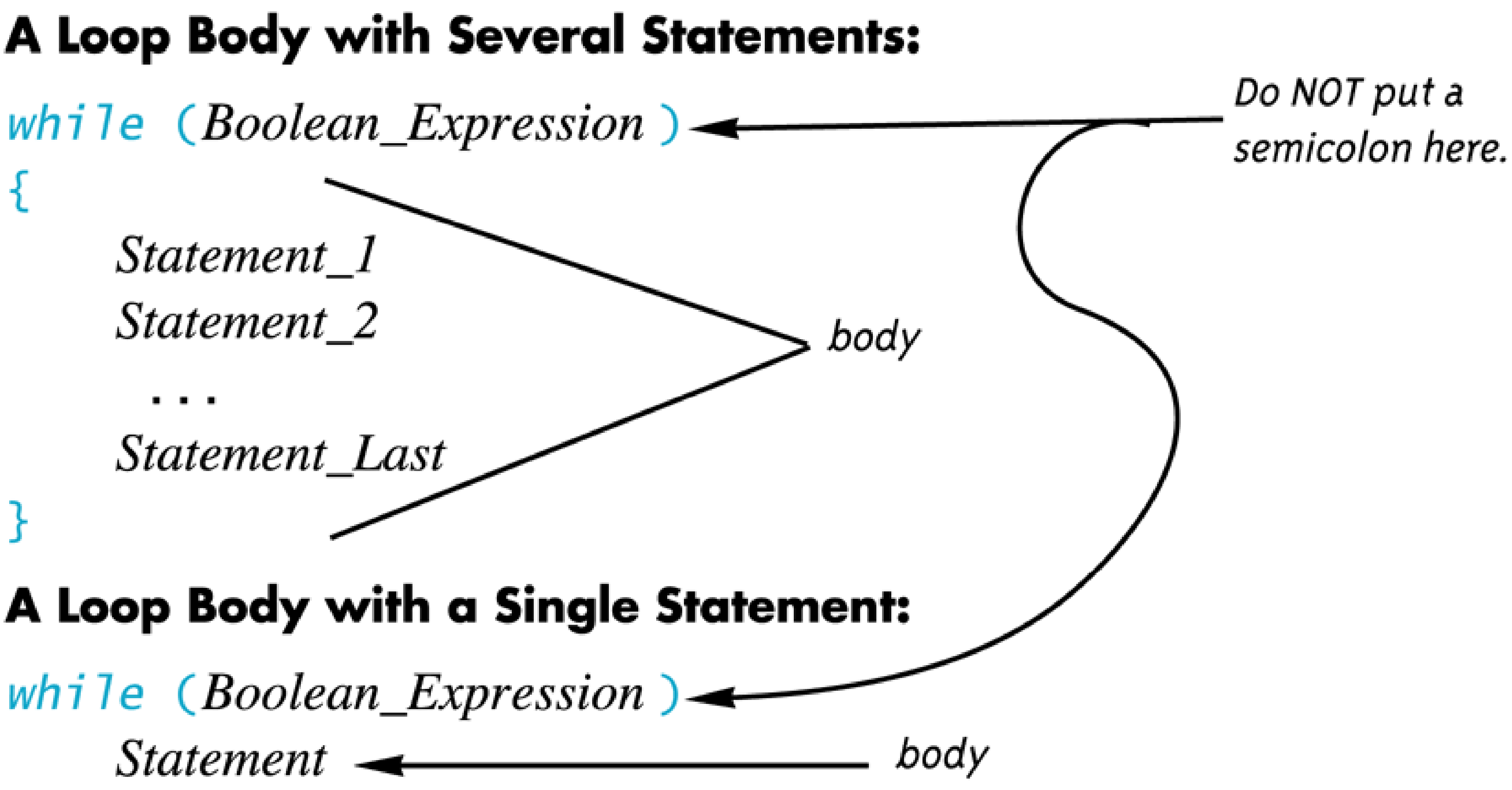
```
How many greetings do you want? 0

That's all!
```

The loop body is executed zero times.

Display 2.12

Back Next

Syntax of the *while* Statement



A Loop Body with Several Statements:

```
while (Boolean_Expression )
{
     Statement_1
     Statement_2
       . . .
     Statement_Last
}
```

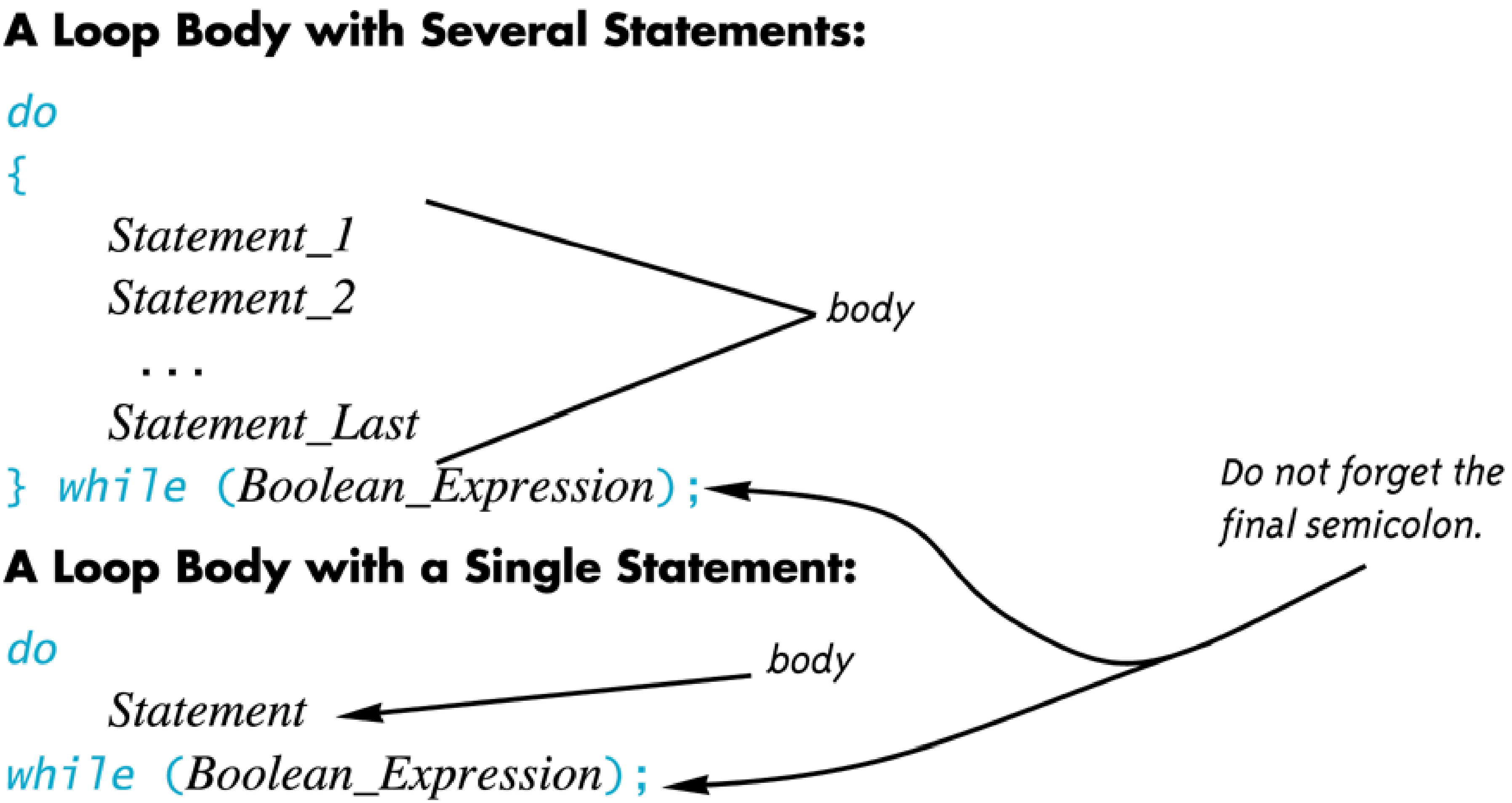
A Loop Body with a Single Statement:

```
while (Boolean_Expression )
     Statement
```

Display 2.13

Back Next

Syntax of the *do-while* Statement



A Loop Body with Several Statements:

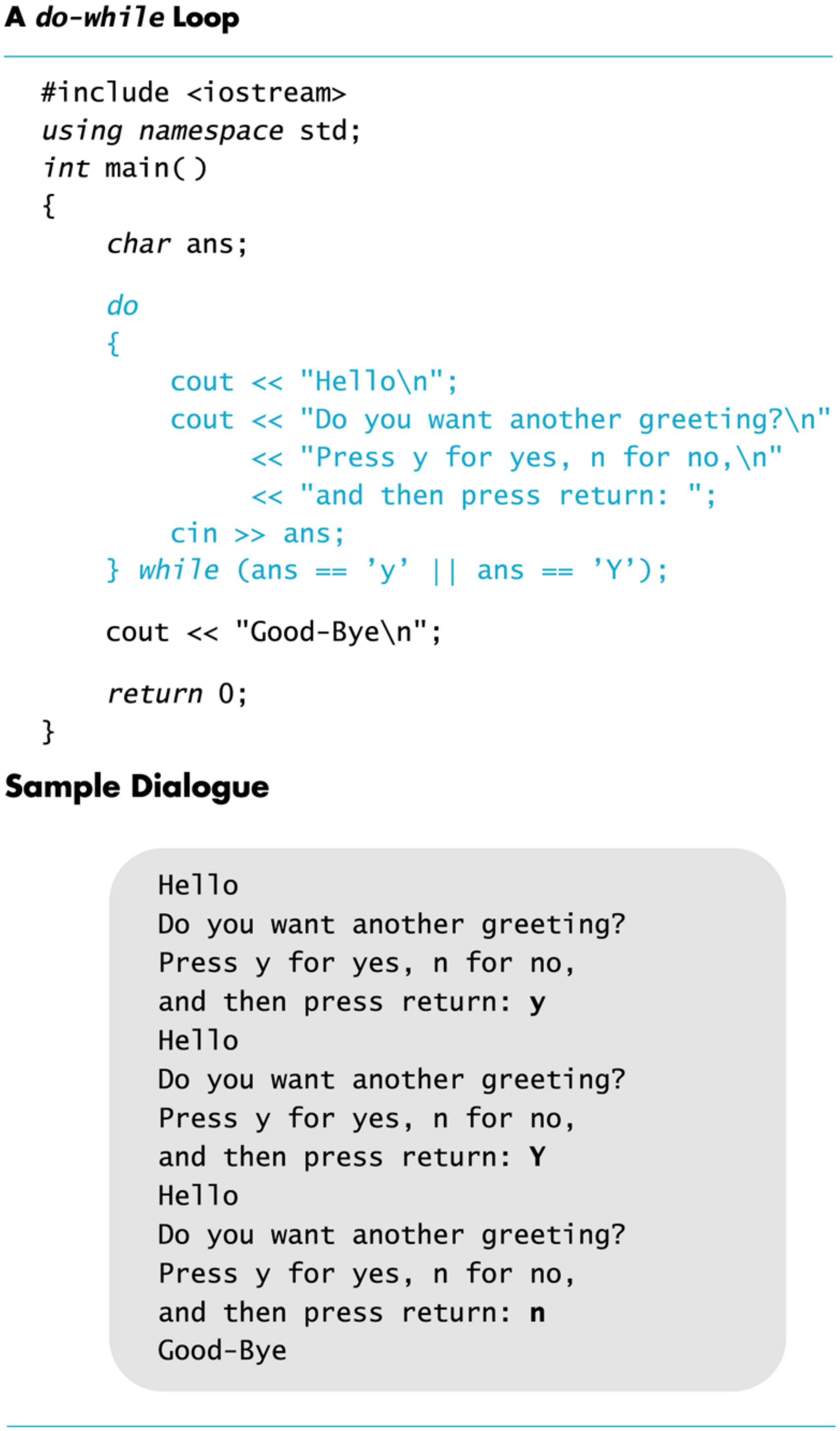
```
do
{
     Statement_1
     Statement_2
       . . .
     Statement_Last
} while (Boolean_Expression);
```

A Loop Body with a Single Statement:

```
do
     Statement
while (Boolean_Expression);
```

Display 2.14

Back Next



A *do-while* Loop

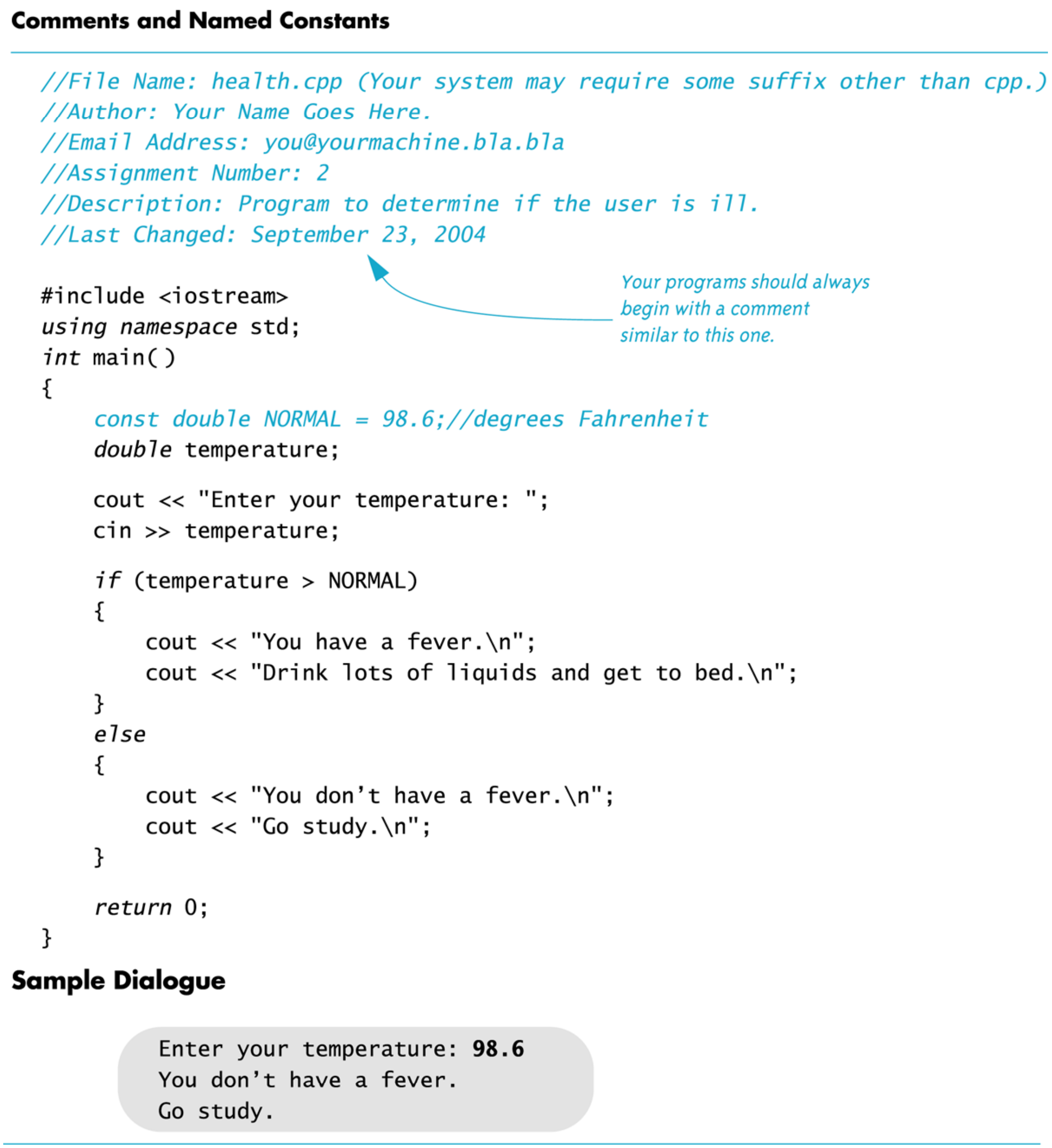
```
#include <iostream>
using namespace std;
int main( )
{
    char ans;

    do
    {
        cout << "Hello\n";
        cout << "Do you want another greeting?\n"
             << "Press y for yes, n for no,\n"
             << "and then press return: ";
        cin >> ans;
    } while (ans == 'y' || ans == 'Y');

    cout << "Good-Bye\n";

    return 0;
}
```

Sample Dialogue

```
Hello
Do you want another greeting?
Press y for yes, n for no,
and then press return: y
Hello
Do you want another greeting?
Press y for yes, n for no,
and then press return: Y
Hello
Do you want another greeting?
Press y for yes, n for no,
and then press return: n
Good-Bye
```

Display 2.16

Back Next



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.

Sample Dialogue

```
Enter your temperature: 98.6
You don’t have a fever.
Go study.
```