

Chapter 4 - Making Decisions

4.1 Relational Operators	4.8 Validating User Input
4.2 The if Statement	4.9 More about blocks & scope
4.3 The if/else Statement	4.10 More about Chars and Strings
4.4 The if/else if Statement	4.11 The Conditional Operator
4.5 Menu-Driven Programs	4.12 The switch Statement
4.6 Nested if Statements	4.13 Enumerated Data Types
4.7 Logical Operators	4.14 Focus on Testing & Debugging

The big idea behind this chapter is

It relates to the previous chapter how ...

The main purpose of this chapter is ...

The key questions are ...

Why:

When:

How:

Why is this material at this point in the class?

You'll know this material when ...

Main assumptions are ...

Opening Thoughts. Write any thoughts or questions you have before reading this material. See if you can find the answers while you read.

Key Ideas. Record major points from the chapter.

[illegible]

List of Figures

1	If statement demo.	4
2	Use if to protect from zero divisor.	5
3	Problem with comparing floats.	5
4	Solution to comparing floats.	6
5	If-else demo.	6
6	if-else chain problem.	7
7	If-else If demo.	8
8	Nested If demo.	9
9	Logical AND demo.	9
10	Logical OR demo.	10
11	Logical NOT demo.	10
12	Numeric menu demo.	11
13	Switch demo.	12
14	Switch fall through demo.	12
15	Fall through demo 2.	13
16	Switch and cin fail demo.	13
17	Shadow variable demo.	14
18	Char test functions demo.	14
19	User input demo.	15
20	Example of how to print enumerations.	16

```
1 // if.cpp -- if demo
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int number;
8     cout << "Enter an integer.\n";
9     cout << "I'll tell you if it's zero.";
10    cin >> number;
11
12    if (number == 0)
13    {
14        cout << "You entered a zero." << endl;
15    }
16    return 0;
17 }
```

Figure 1: §4.2 If statement demo.
Source file: if.cpp

```

1 // 4-4.cpp -- protect from zero divisor
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     double num1, num2, quotient;
7
8     // get the numbers
9     cout << "Enter two numbers: ";
10    cin >> num1 >> num2;
11
12    // Perform division if num2 isn't zero
13    if (num2 != 0) {
14        quotient = num1 / num2;
15        cout << "The quotient of " << num1 << " divided by "
16             << num2 << " is " << quotient << endl;
17    } else {
18        cout << "Division by zero is not possible.\n";
19        cout << "Please rerun program." << endl;
20    }
21    return 0;
22 }

```

Figure 2: §4.2 Use if to protect from zero divisor. Source file: 4-4.cpp

```

1 // 4-5.cpp -- problem with floats
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     double result = .666667 * 6.0;
7     // 2/3 of 6 s.b. 4 - prints correctly
8     cout << "printed result = " << result << endl;
9
10    // internally, it is NOT 4
11    if (result == 4.0) {
12        cout << "internal result = 4.0" << endl;
13    } else {
14        cout << "internal result does not = 4.0!" << endl;
15    }
16
17    return 0;
18 }

```

Figure 3: §4.2 Problem with comparing floats. Source file: 4-5.cpp

```

1 // 4-6.cpp -- comparing floats
2 #include <iostream>
3 #include <cmath>      // for abs()
4 using namespace std;
5
6 int main() {
7     double result = .666667 * 6.0;
8     // 2/3 of 6 s.b. 4 - prints correctly
9     cout << "printed result = " << result << endl;
10
11     // internally, it is NOT 4
12     // so, we test to see if it's close to 4
13     if (abs(result - 4.0) < .0001) {
14         cout << "internal result = 4.0" << endl;
15     } else {
16         cout << "internal result does not = 4.0!" << endl;
17     }
18
19     return 0;
20 }

```

Figure 4: §4.2 Solution to comparing floats. Source file: 4-6.cpp

```

1 // 4-3.cpp      -- if/else demo
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int number;
8     cout << " Enter an integer\n";
9     cout << "I'll tell you if it's\n";
10    cout << "odd or even. ";
11    cin >> number;
12    if (number % 2 == 0)
13        cout << number << " is even.\n";
14    else
15        cout << number << " is odd.\n";
16    return 0;
17 }

```

Figure 5: §4.3 If-else demo. Source file: 4-3.cpp

```

1 // 4-8mod.cpp -- if/else chain problem
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     const int MIN_A = 90,
7             MIN_B = 80,
8             MIN_C = 70,
9             MIN_D = 60;
10    int testScore;
11    char grade;
12
13    cout << "Enter your score: ";
14    cin >> testScore;
15
16    if (testScore >= MIN_A)
17        grade = 'A';
18    else {
19        if (testScore >= MIN_B)
20            grade = 'B';
21        else {
22            if (testScore >= MIN_C)
23                grade = 'C';
24            else {
25                if (testScore >= MIN_D)
26                    grade = 'D';
27                else {
28                    if (testScore >= 0)
29                        grade = 'F';
30                }
31            }
32        }
33    }
34
35    cout << "Your grade is a " << grade << endl;
36
37    return 0;
38 }

```

Figure 6: §4.3 if-else chain problem.
Source file: 4-8mod.cpp

```

1 // 4-9.cpp -- if/else if + trailing else
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     const int    MIN_A = 90,
7                 MIN_B = 80,
8                 MIN_C = 70,
9                 MIN_D = 60,
10                MIN_POSSIBLE = 0;
11     int testScore ;
12     char grade;
13     bool goodScore = true;
14
15     cout << "Enter your score: ";
16     cin >> testScore;
17
18     if (testScore >= MIN_A)
19         grade = 'A';
20
21     else if (testScore >= MIN_B)
22         grade = 'B';
23
24     else if (testScore >= MIN_C)
25         grade = 'C';
26
27     else if (testScore >= MIN_D)
28         grade = 'D';
29
30     else if (testScore >= MIN_POSSIBLE)
31         grade = 'F';
32
33     else
34         goodScore = false;
35
36     if (goodScore)
37         cout << "Your grade is a " << grade << endl;
38     else
39         cout << "Score cannot be negative\n";
40
41     return 0;
42 }

```

Figure 7: §4.4 If-else If demo. Source file: 4-9.cpp


```

1 // 4-11.cpp -- Nested If Demo demo
2 #include <iostream>
3 using namespace std;
4
5
6 int main() {
7     char    employed,      // y or n
8           recentGrad; // y or n
9
10    cout << "Are you employed? (y/n) ";
11    cin >> employed;
12    cout << "Are you a college graduate? (y/n) ";
13    cin >> recentGrad;
14
15    if (recentGrad == 'y')
16    {
17        if (employed == 'y')
18        {
19            cout << "You qualify for the special rate.\n";
20        }
21        else
22        {
23            // grad, but not employed
24            cout << "You need a job to qualify.\n";
25        }
26    }
27    else
28    {
29        // employed, but not a grad
30        cout << "You qualify for the general rate.\n";
31    }
32    return 0;
}

```

Figure 8: §4.6 Nested If demo. Source file: 4-11.cpp

```

1 // 4-12.cpp -- Logical AND demo
2 #include <iostream>
3 using namespace std;
4
5
6 int main() {
7     char    employed,      // y or n
8           recentGrad; // y or n
9
10    cout << "Are you employed? (y/n) ";
11    cin >> employed;
12    cout << "Are you a college graduate? (y/n) ";
13    cin >> recentGrad;
14
15    if (recentGrad == 'y' && employed == 'y')
16    {
17        cout << "You qualify for the special rate.\n";
18    }
19    else
20    {
21        cout << "You don't qualify for the special rate.\n";
22    }
23
24    return 0;
25 }

```

Figure 9: §4.7 Logical AND demo. Source file: 4-12.cpp

```

1 // 4-13.cpp -- Logical AND demo
2 #include <iostream>
3 using namespace std;
4
5
6 int main() {
7     char    employed,      // y or n
8           recentGrad;      // y or n
9
10    cout << "Are you employed? (y/n) ";
11    cin >> employed;
12    cout << "Are you a college graduate? (y/n) ";
13    cin >> recentGrad;
14
15    if (recentGrad == 'y' || employed == 'y')
16    {
17        cout << "You may qualify for the special rate.\n";
18    }
19    else
20    {
21        cout << "You don't qualify for a loan.\n";
22    }
23
24    return 0;
25 }

```

Figure 10: §4.7 Logical OR demo.
Source file: 4-13.cpp

```

1 // 4-14.cpp -- Logical NOT demo
2 #include <iostream>
3 using namespace std;
4
5
6 int main() {
7     char    employed,      // y or n
8           recentGrad;      // y or n
9
10    cout << "Are you employed? (y/n) ";
11    cin >> employed;
12    cout << "Are you a college graduate? (y/n) ";
13    cin >> recentGrad;
14
15    if (!(recentGrad == 'y' || employed == 'y'))
16    {
17        cout << "You don't qualify for a loan.\n";
18    }
19    else
20    {
21        cout << "You may qualify for the special rate.\n";
22    }
23
24    return 0;
25 }

```

Figure 11: §4.7 Logical NOT demo.
Source file: 4-14.cpp

```

1 // 4-10.cpp -- Menu demo
2 #include <iostream>
3 #include <iomanip>
4 using namespace std;
5
6
7 int main() {
8     const double    ADULT_RATE = 120.0,
9                     CHILD_RATE = 60.0,
10                    SENIOR_RATE = 100.0;
11     int choice;
12     int months;
13     double charges;
14
15     // Display Menu
16     cout << "\tHealth Club Membership Prices\n\n";
17     cout << "1. Standard Adult Membership\n";
18     cout << "2. Child Membership\n";
19     cout << "3. Senior Membership\n";
20     cout << "4. Quit Program\n\n";
21     cout << "Enter your choice: ";
22     cin >> choice;
23
24     cout << fixed << showpoint << setprecision(2);
25
26     // process menu options
27     if (choice == 1)
28     {   cout << "For how many months? ";
29         cin >> months;
30         charges = months * ADULT_RATE;
31         cout << "\nThe total charges are $" << charges << endl;
32     }
33     else if (choice == 2)
34     {   cout << "For how many months? ";
35         cin >> months;
36         charges = months * CHILD_RATE;
37         cout << "\nThe total charges are $" << charges << endl;
38     }
39     else if (choice == 3)
40     {   cout << "For how many months? ";
41         cin >> months;
42         charges = months * SENIOR_RATE;
43         cout << "\nThe total charges are $" << charges << endl;
44     }
45     else if (choice != 4)
46     {   cout << "\nValid choices are 1 - 4" << endl;
47         cout << "Terminating program." << endl;
48     }
49
50     return 0;
51 }

```

Figure 12: §4.5 Numeric menu demo.
Source file: 4-10.cpp

```

1 // 4-23.cpp -- switch demo
2 #include <iostream>
3 using namespace std;
4
5
6 int main() {
7     char choice;
8
9     cout << "Enter A, B, C: ";
10    cin >> choice;
11
12    switch (choice) {
13        case 'A':    cout << "You entered A.\n";
14                    break;
15        case 'B':    cout << "You entered B.\n";
16                    break;
17        case 'C':    cout << "You entered A.\n";
18                    break;
19        default:     cout << "Incorrect entry!\n";
20    }
21
22    return 0;
23 }

```

Figure 13: §4.12 Switch demo. Source file: 4-23.cpp

```

1 // 4-25.cpp -- switch fall through demo
2 #include <iostream>
3 using namespace std;
4
5
6 int main() {
7     char choice;
8
9     cout << "Enter A, B, C: ";
10    cin >> choice;
11
12    switch (choice) {
13        case 'a':
14        case 'A':    cout << "You entered A.\n";
15                    break;
16        case 'b':
17        case 'B':    cout << "You entered B.\n";
18                    break;
19        case 'c':
20        case 'C':    cout << "You entered C.\n";
21                    break;
22        default:     cout << "Incorrect entry!\n";
23    }
24
25    return 0;
26 }

```

Figure 14: §4.12 Switch fall through demo. Source file: 4-25a.cpp

```

1 // 4-25.cpp -- switch fall through demo 2
2 #include <iostream>
3 using namespace std;
4
5
6 int main() {
7     int model;
8
9     cout << "Our TV's come in 3 maodels: The 100, 200 and 300.\n";
10    cout << "Which would you like? ";
11    cin >> model;
12
13    cout << "\nThat model has the following features:\n";
14    switch (model) {
15        case 300: cout << "Built in DVR.\n";
16        case 200: cout << "Hi Def picture.\n";
17        case 100: cout << "42\" LCD flat screen.\n";
18                break;
19        default: cout << "We don't have that model." << endl;
20    }
21
22    return 0;
23 }

```

Figure 15: §4.12 Fall through demo 2.
Source file: 4-25.cpp

```

1 // cinSwitch.cpp -- cin and switch statements
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     int cmd = 0;
7
8     cout << "Enter your command: ";
9     switch (cin >> cmd) {
10        case 1: cout << "Something" << endl;
11        case 2: cout << "Something else" << endl;
12    }
13
14    // This does not work.
15    // cin returns a ifstream object, not an int.
16    // Switch only recognizes int or enumerations.
17
18    return 0;
19 }

```

Figure 16: §4.12 Switch and cin fail demo. Source file: cinSwitch.cpp

```

1 // 4-18.cpp -- shadow variables
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int number;
8
9     cout << "Enter a number greater than 0: ";
10    cin >> number;
11
12    if (number > 0)
13    { // create nested scope
14        int number;
15
16        cout << "Enter another number: ";
17        cin >> number;
18        cout << "The second number is: " << number << endl;
19    }
20    cout << "The first number is: " << number << endl;
21    return 0;
22 }

```

Figure 17: §4.9Shadow variable demo.
Source file: 4-18.cpp

```

1 // 4-21.cpp -- char test demo
2 #include <iostream>
3 #include <cctype> // use for char testing functions
4 using namespace std;
5
6 int main() {
7     char input;
8     cout << "Enter a character: ";
9     cin.get(input);
10
11     cout << "You entered " << input << ", ";
12     cout << "with an ASCII code of " << static_cast<int>(input) << endl;
13
14     if (isalpha(input))
15         cout << "It's alphabetic.\n";
16     if (isdigit(input))
17         cout << "It's a digit.\n";
18     if (islower(input))
19         cout << "It's in lowercase.\n";
20     if (isupper(input))
21         cout << "It's uppercase.\n";
22     if (ispunct(input))
23         cout << "It's punctuation.\n";
24     if (isspace(input))
25         cout << "It's a whitespace character.\n";
26
27     return 0;
28 }

```

Figure 18: §4.10 Char test functions
demo. Source file: 4-21.cpp

```

1
2
3
4
5 // 4-16.cpp -- User Input demo
6 #include <iostream>
7 using namespace std;
8
9
10 int main() {
11     const int    A_SCORE = 90,
12                 B_SCORE = 80,
13                 C_SCORE = 70,
14                 D_SCORE = 60,
15                 MIN_SCORE = 0,
16                 MAX_SCORE = 100;
17     int testScore;
18
19     cout << "Enter your test score and I will tell you your grade: ";
20     cin >> testScore;
21
22     // is it valid?
23     if (testScore >= MIN_SCORE && testScore <= MAX_SCORE) {
24         if (testScore >= A_SCORE)
25             cout << "Your grade is an A.\n";
26         else if (testScore >= B_SCORE)
27             cout << "Your grade is a B.\n";
28         else if (testScore >= C_SCORE)
29             cout << "Your grade is a C.\n";
30         else if (testScore >= D_SCORE)
31             cout << "Your grade is a D.\n";
32         else
33             cout << "Your score is an F.\n";
34     }
35     else
36     { // invalid score was entered
37         cout << "That was an invalid score. Rerun the program using\n";
38         cout << "a score between " << MIN_SCORE << " and " << MAX_SCORE;
39         cout << endl;
40     }
41
42     return 0;
43 }

```

Figure 19: §4.8 User input demo.
Source file: 4-16.cpp

```

1 // enumEx.cpp -- enumeration example
2 #include <iostream>
3 using namespace std;
4
5 enum colors {BROWN, BLUE, RED, GREEN};
6
7 void PrintEnum(colors);
8
9 int main()
10 {
11     PrintEnum(BLUE);
12     return 0;
13 }
14
15 void PrintEnum(colors c) {
16     switch (c)
17     {
18         case BROWN: cout << "Brown";
19                     break;
20         case BLUE:   cout << "Blue";
21                     break;
22         case RED:    cout << "Red";
23                     break;
24         case GREEN:  cout << "Green";
25                     break;
26         default:     cout << "PrintEnum failure.";
27     }
28 }

```

Figure 20: §4.13 Example of how to print enumerations. Source file: enumEx.cpp