Chapter 4 - Making Decisions

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 ...

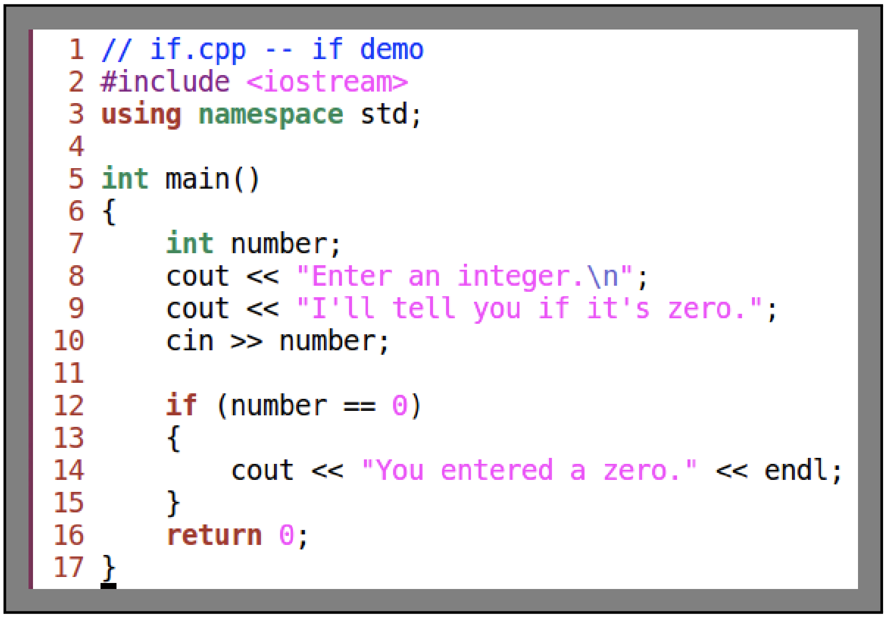
Opening Thoughts. Write any thoughts or questions you have before reading this material. See if you can find the answers while you read.

Key Ideas. Record major points from the chapter.

Idea	Notes

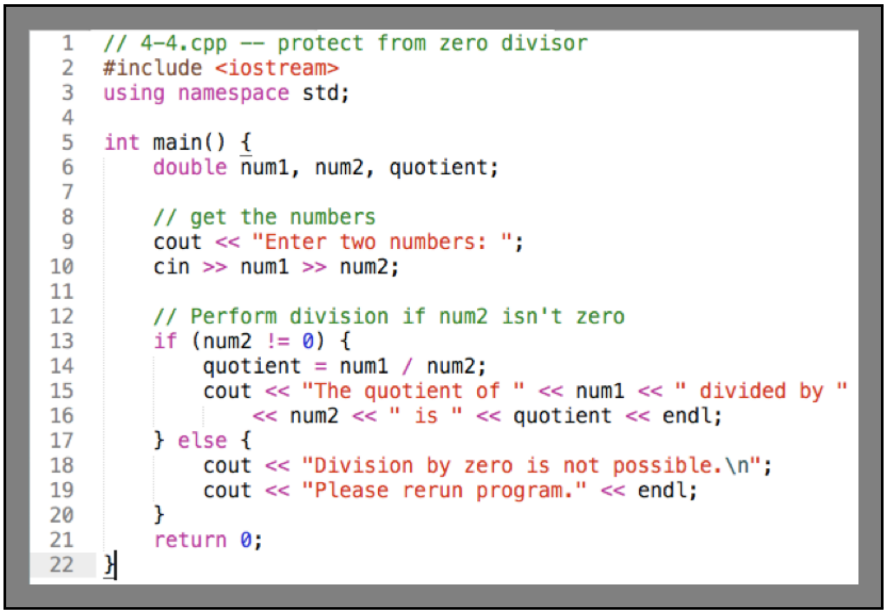
List of Figures

Demonstration Code.



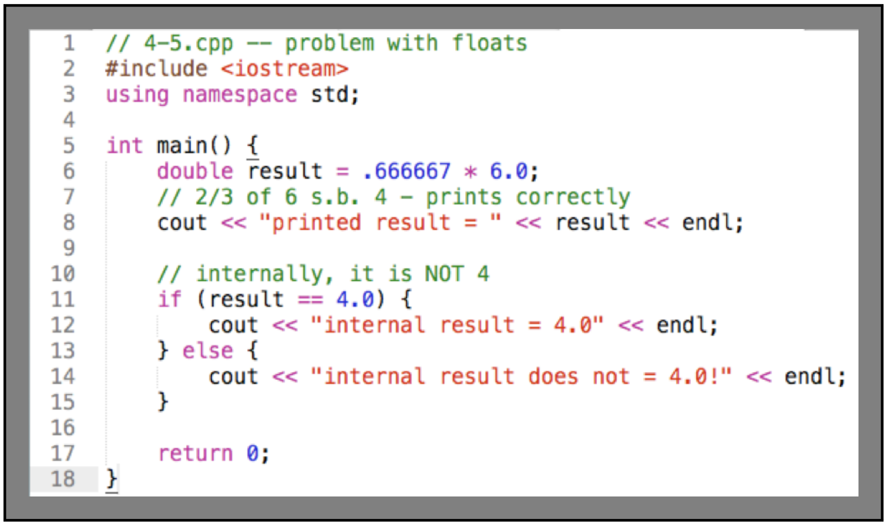
```cpp
// if.cpp -- if demo
#include <iostream>
using namespace std;

int main()
{
    int number;
    cout << "Enter an integer.\n";
    cout << "I'll tell you if it's zero.";
    cin >> number;

    if (number == 0)
    {
        cout << "You entered a zero." << endl;
    }
    return 0;
}
```

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



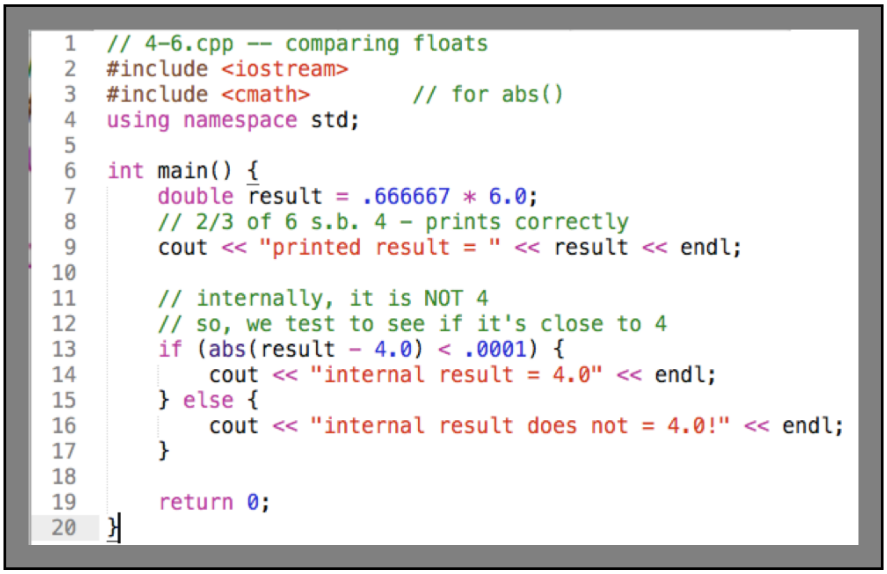
```cpp
// 4-4.cpp -- protect from zero divisor
#include <iostream>
using namespace std;

int main() {
    double num1, num2, quotient;

    // get the numbers
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;

    // Perform division if num2 isn't zero
    if (num2 != 0) {
        quotient = num1 / num2;
        cout << "The quotient of " << num1 << " divided by "
            << num2 << " is " << quotient << endl;
    } else {
        cout << "Division by zero is not possible.\n";
        cout << "Please rerun program." << endl;
    }
    return 0;
}
```

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



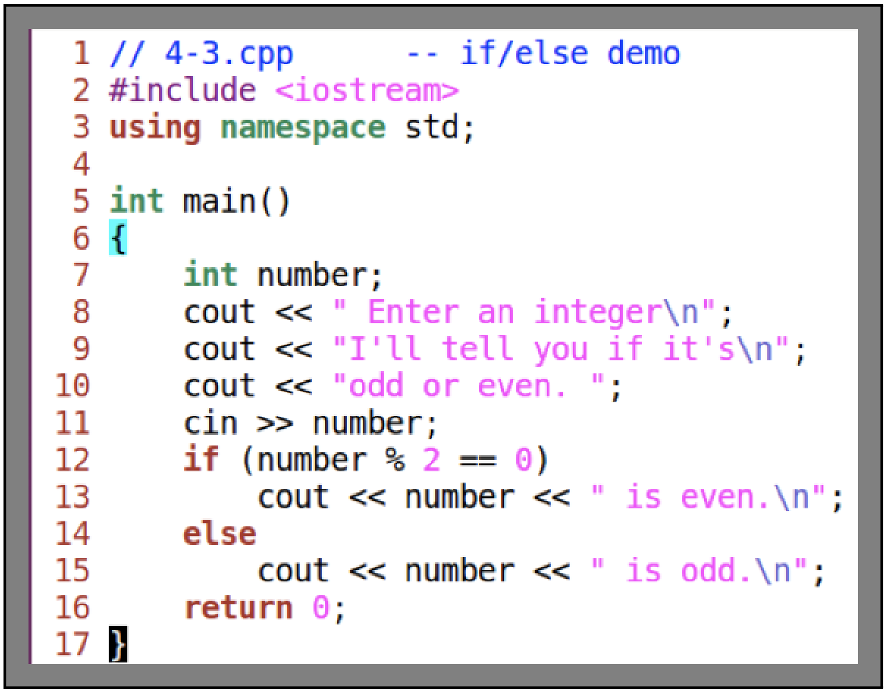
```cpp
// 4-5.cpp -- problem with floats
#include <iostream>
using namespace std;

int main() {
    double result = .666667 * 6.0;
    // 2/3 of 6 s.b. 4 - prints correctly
    cout << "printed result = " << result << endl;

    // internally, it is NOT 4
    if (result == 4.0) {
        cout << "internal result = 4.0" << endl;
    } else {
        cout << "internal result does not = 4.0!" << endl;
    }

    return 0;
}
```

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



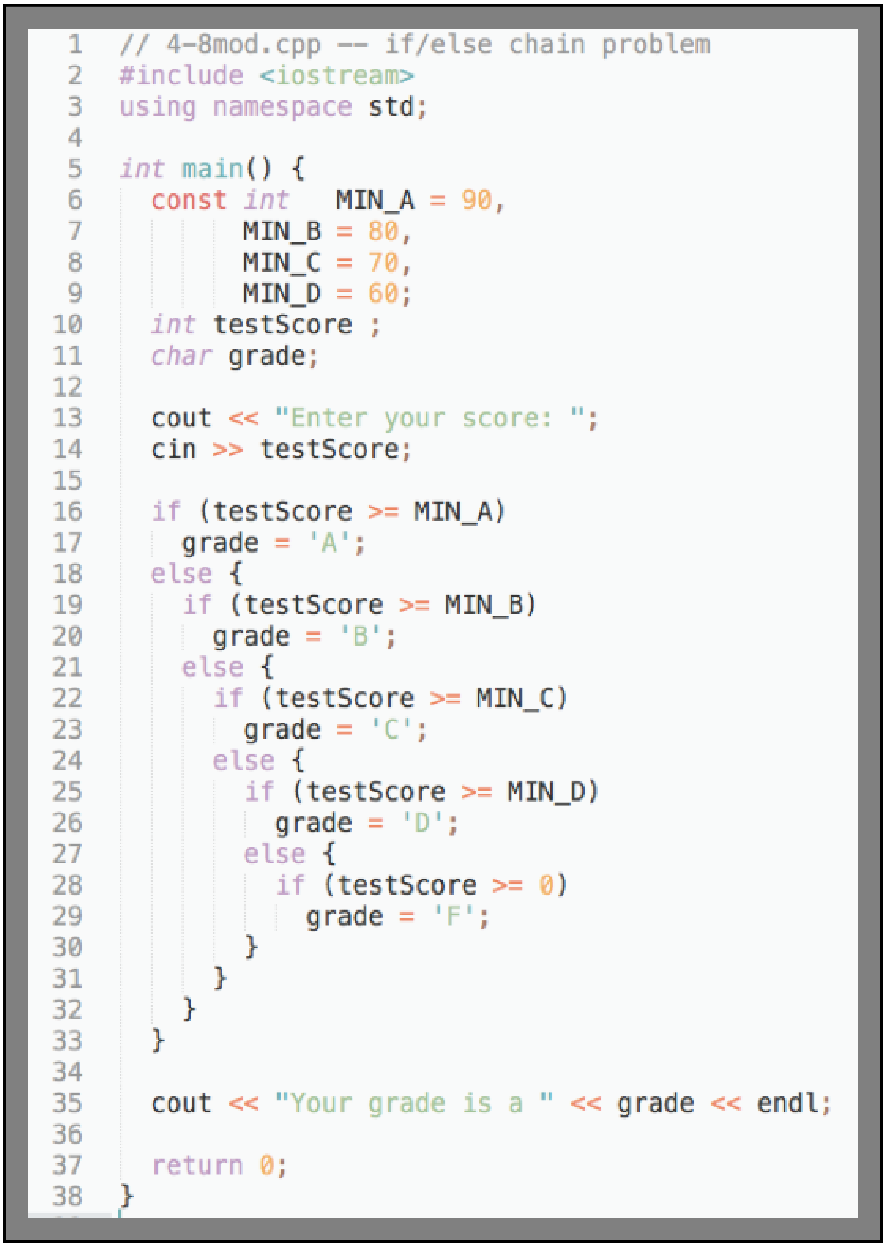
```cpp
// 4-6.cpp -- comparing floats
#include <iostream>
#include <cmath>        // for abs()
using namespace std;

int main() {
    double result = .666667 * 6.0;
    // 2/3 of 6 s.b. 4 - prints correctly
    cout << "printed result = " << result << endl;

    // internally, it is NOT 4
    // so, we test to see if it's close to 4
    if (abs(result - 4.0) < .0001) {
        cout << "internal result = 4.0" << endl;
    } else {
        cout << "internal result does not = 4.0!" << endl;
    }

    return 0;
}
```

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



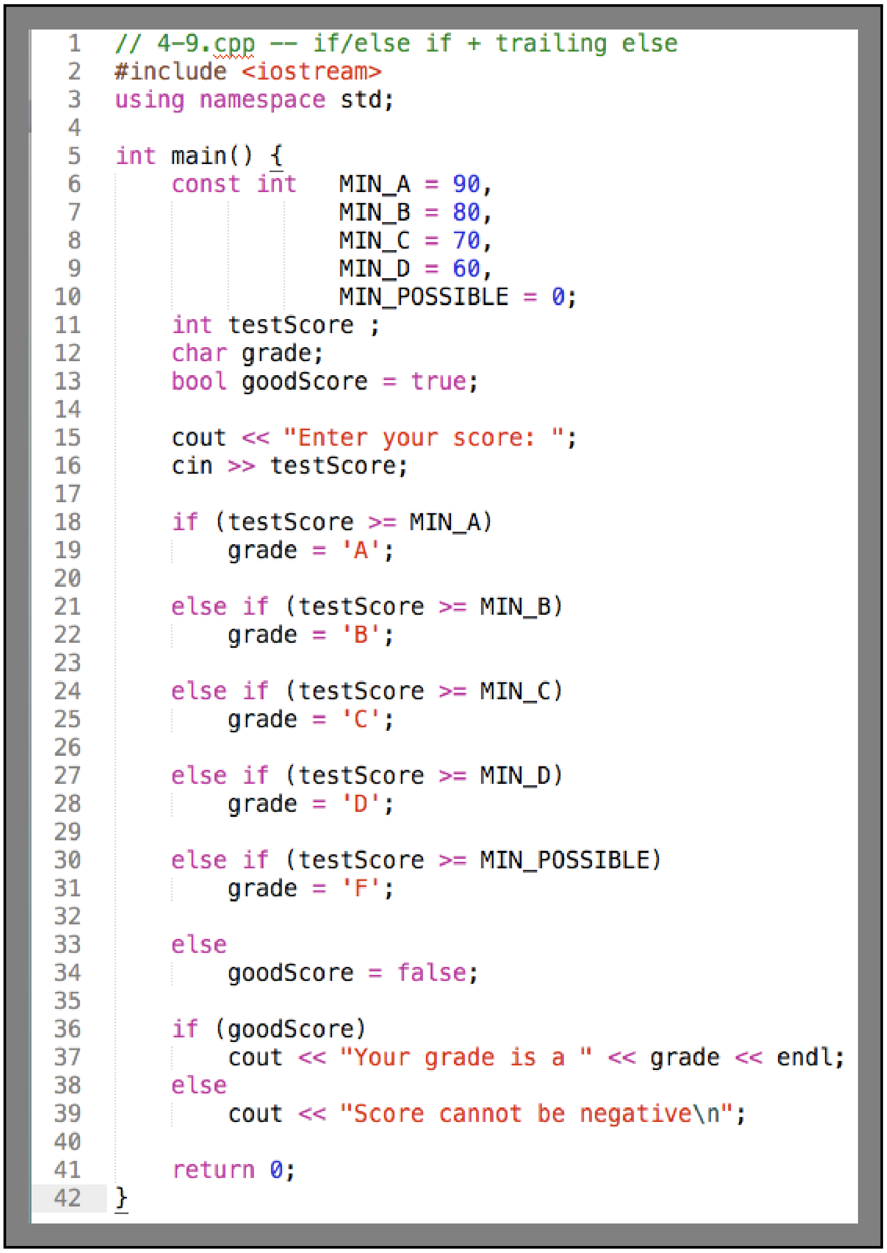
```cpp
// 4-3.cpp      -- if/else demo
#include <iostream>
using namespace std;

int main()
{
    int number;
    cout << " Enter an integer\n";
    cout << "I'll tell you if it's\n";
    cout << "odd or even. ";
    cin >> number;
    if (number % 2 == 0)
        cout << number << " is even.\n";
    else
        cout << number << " is odd.\n";
    return 0;
}
```

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



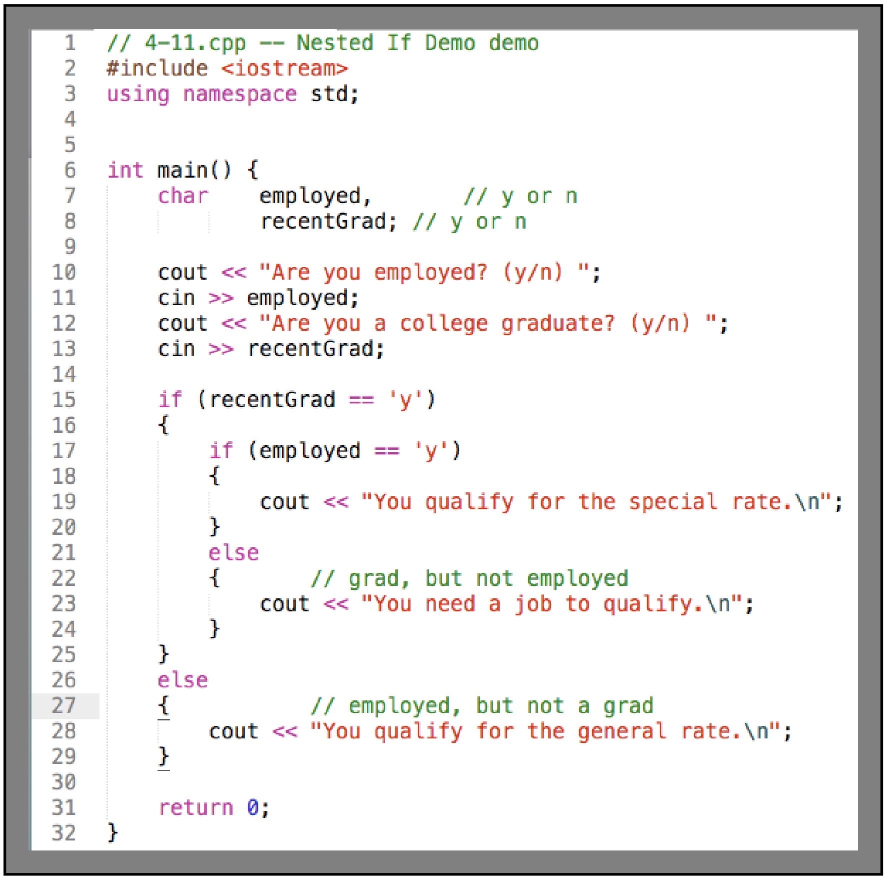
```cpp
// 4-8mod.cpp -- if/else chain problem
#include <iostream>
using namespace std;

int main() {
  const int   MIN_A = 90,
        MIN_B = 80,
        MIN_C = 70,
        MIN_D = 60;
  int testScore ;
  char grade;

  cout << "Enter your score: ";
  cin >> testScore;

  if (testScore >= MIN_A)
    grade = 'A';
  else {
    if (testScore >= MIN_B)
      grade = 'B';
    else {
      if (testScore >= MIN_C)
        grade = 'C';
      else {
        if (testScore >= MIN_D)
          grade = 'D';
        else {
          if (testScore >= 0)
            grade = 'F';
        }
      }
    }
  }

  cout << "Your grade is a " << grade << endl;

  return 0;
}
```

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



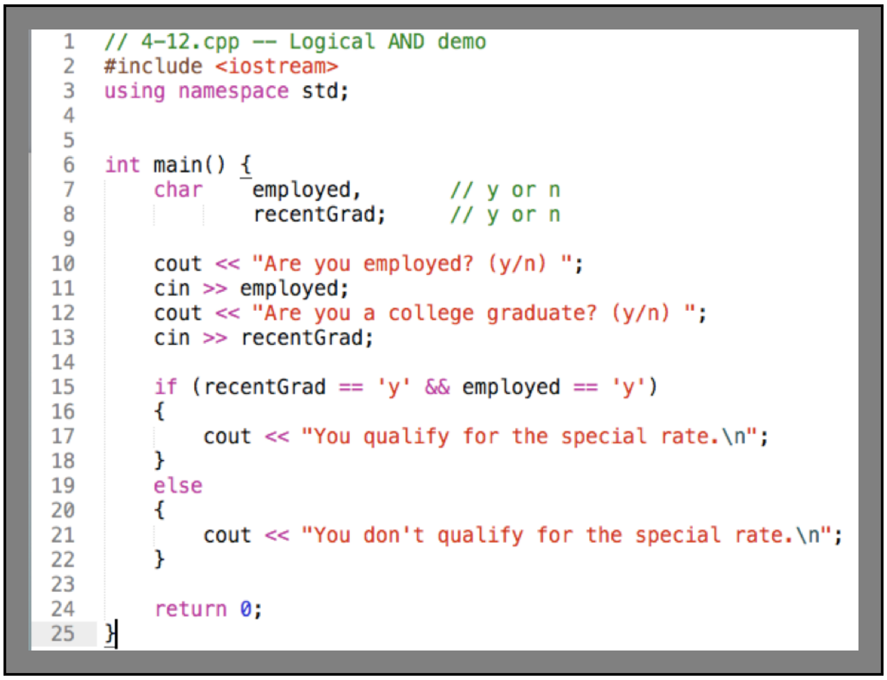
```cpp
// 4-9.cpp -- if/else if + trailing else
#include <iostream>
using namespace std;

int main() {
    const int   MIN_A = 90,
                MIN_B = 80,
                MIN_C = 70,
                MIN_D = 60,
                MIN_POSSIBLE = 0;
    int testScore ;
    char grade;
    bool goodScore = true;

    cout << "Enter your score: ";
    cin >> testScore;

    if (testScore >= MIN_A)
        grade = 'A';

    else if (testScore >= MIN_B)
        grade = 'B';

    else if (testScore >= MIN_C)
        grade = 'C';

    else if (testScore >= MIN_D)
        grade = 'D';

    else if (testScore >= MIN_POSSIBLE)
        grade = 'F';

    else
        goodScore = false;

    if (goodScore)
        cout << "Your grade is a " << grade << endl;
    else
        cout << "Score cannot be negative\n";

    return 0;
}
```

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



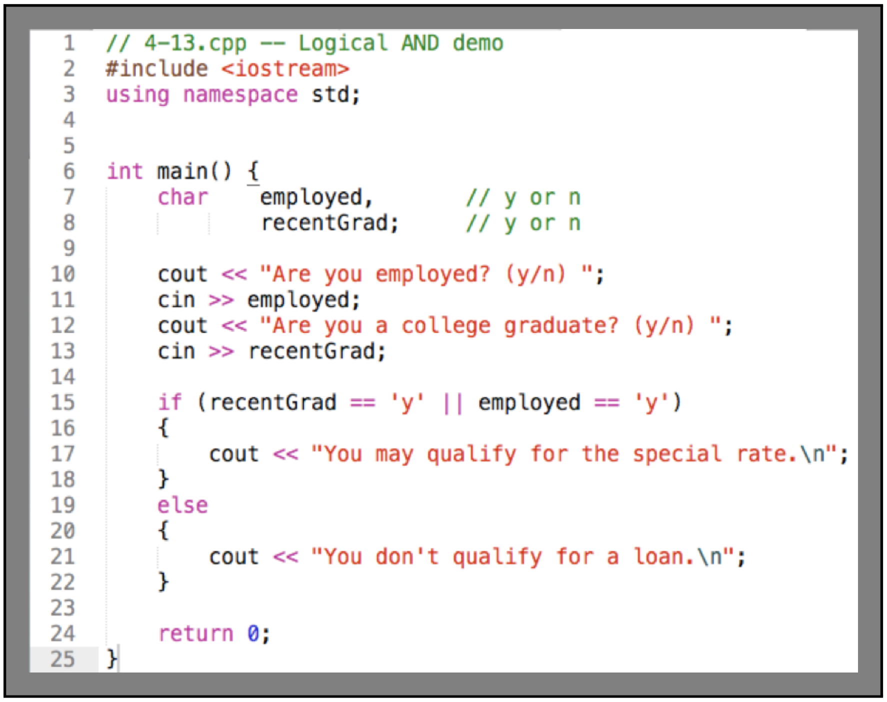
```cpp
// 4-11.cpp -- Nested If Demo demo
#include <iostream>
using namespace std;


int main() {
    char    employed,       // y or n
            recentGrad; // y or n

    cout << "Are you employed? (y/n) ";
    cin >> employed;
    cout << "Are you a college graduate? (y/n) ";
    cin >> recentGrad;

    if (recentGrad == 'y')
    {
        if (employed == 'y')
        {
            cout << "You qualify for the special rate.\n";
        }
        else
        {       // grad, but not employed
            cout << "You need a job to qualify.\n";
        }
    }
    else
    {           // employed, but not a grad
        cout << "You qualify for the general rate.\n";
    }

    return 0;
}
```

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



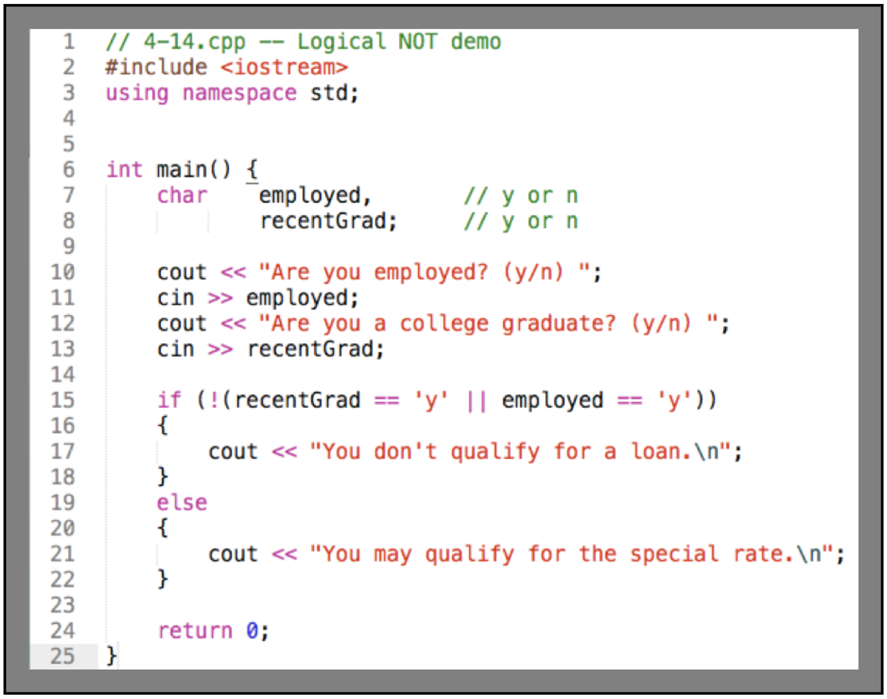
```cpp
// 4-12.cpp -- Logical AND demo
#include <iostream>
using namespace std;


int main() {
    char    employed,       // y or n
            recentGrad;     // y or n

    cout << "Are you employed? (y/n) ";
    cin >> employed;
    cout << "Are you a college graduate? (y/n) ";
    cin >> recentGrad;

    if (recentGrad == 'y' && employed == 'y')
    {
        cout << "You qualify for the special rate.\n";
    }
    else
    {
        cout << "You don't qualify for the special rate.\n";
    }

    return 0;
}
```

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



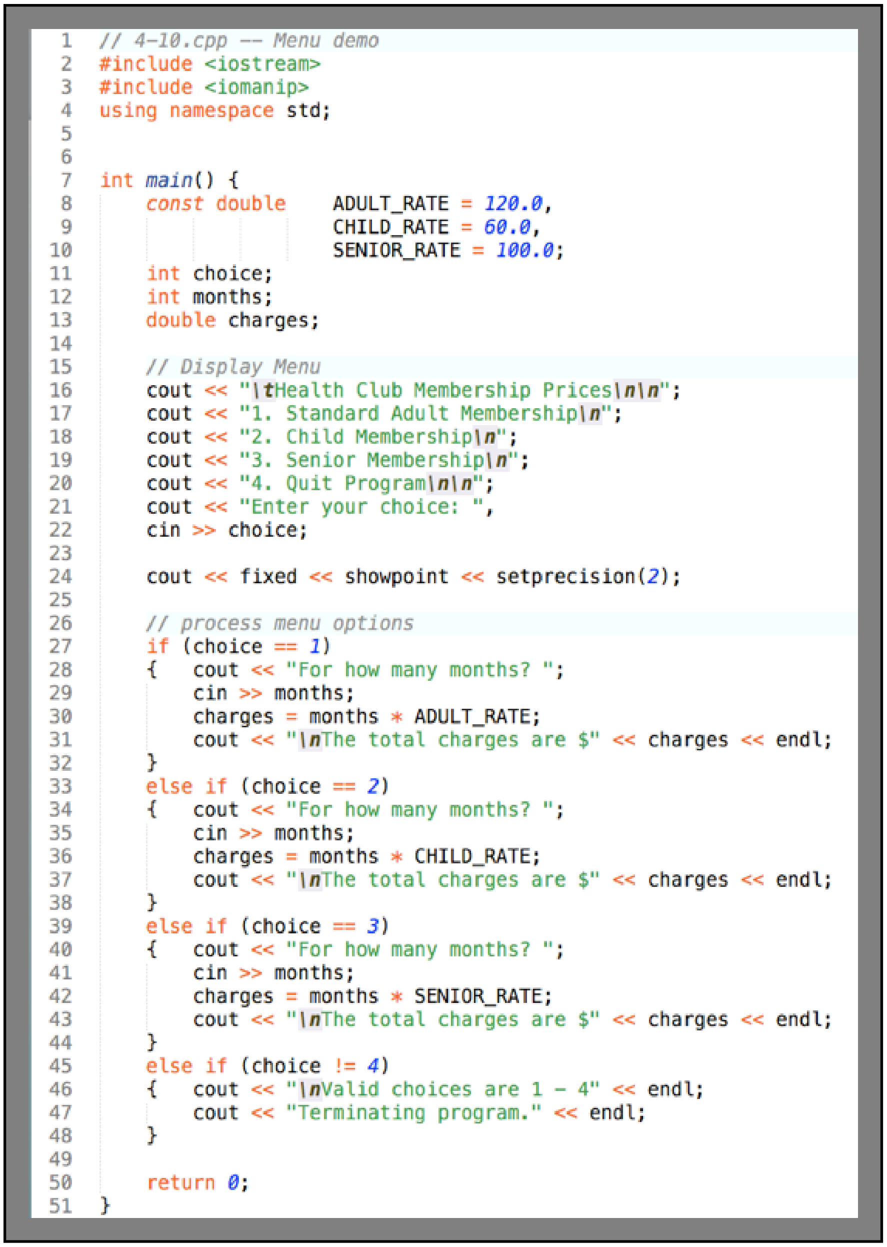
```cpp
// 4-13.cpp -- Logical AND demo
#include <iostream>
using namespace std;


int main() {
    char    employed,       // y or n
            recentGrad;     // y or n

    cout << "Are you employed? (y/n) ";
    cin >> employed;
    cout << "Are you a college graduate? (y/n) ";
    cin >> recentGrad;

    if (recentGrad == 'y' || employed == 'y')
    {
        cout << "You may qualify for the special rate.\n";
    }
    else
    {
        cout << "You don't qualify for a loan.\n";
    }

    return 0;
}
```

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



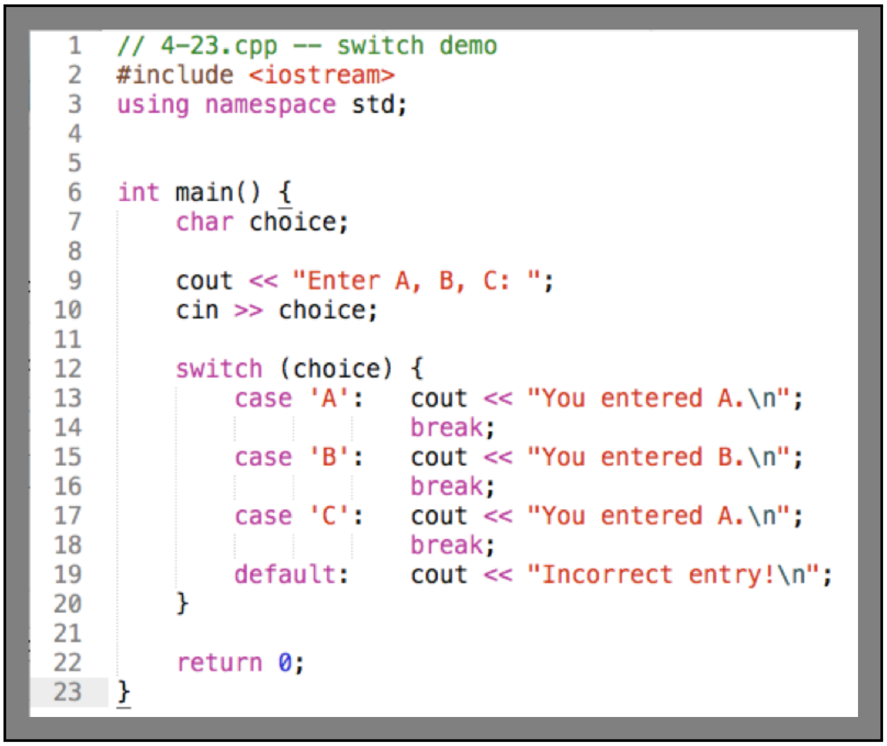
```cpp
// 4-14.cpp -- Logical NOT demo
#include <iostream>
using namespace std;


int main() {
    char    employed,       // y or n
            recentGrad;     // y or n

    cout << "Are you employed? (y/n) ";
    cin >> employed;
    cout << "Are you a college graduate? (y/n) ";
    cin >> recentGrad;

    if (!(recentGrad == 'y' || employed == 'y'))
    {
        cout << "You don't qualify for a loan.\n";
    }
    else
    {
        cout << "You may qualify for the special rate.\n";
    }

    return 0;
}
```

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



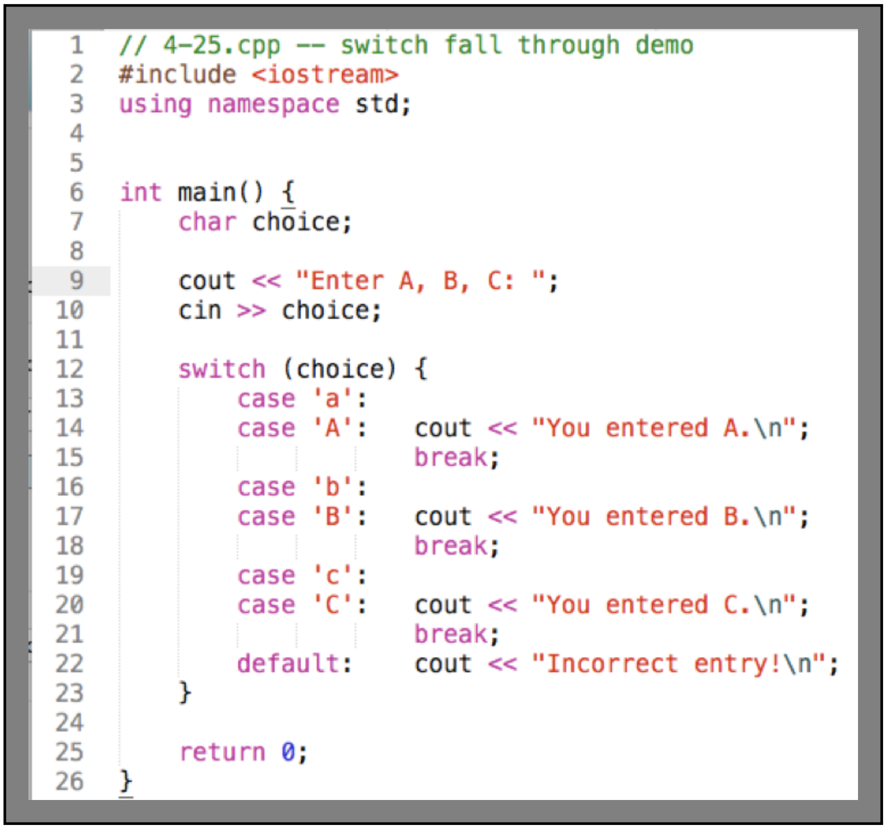
```cpp
// 4-10.cpp -- Menu demo
#include <iostream>
#include <iomanip>
using namespace std;


int main() {
    const double    ADULT_RATE = 120.0,
                    CHILD_RATE = 60.0,
                    SENIOR_RATE = 100.0;
    int choice;
    int months;
    double charges;

    // Display Menu
    cout << "\tHealth Club Membership Prices\n\n";
    cout << "1. Standard Adult Membership\n";
    cout << "2. Child Membership\n";
    cout << "3. Senior Membership\n";
    cout << "4. Quit Program\n\n";
    cout << "Enter your choice: ",
    cin >> choice;

    cout << fixed << showpoint << setprecision(2);

    // process menu options
    if (choice == 1)
    {   cout << "For how many months? ";
        cin >> months;
        charges = months * ADULT_RATE;
        cout << "\nThe total charges are $" << charges << endl;
    }
    else if (choice == 2)
    {   cout << "For how many months? ";
        cin >> months;
        charges = months * CHILD_RATE;
        cout << "\nThe total charges are $" << charges << endl;
    }
    else if (choice == 3)
    {   cout << "For how many months? ";
        cin >> months;
        charges = months * SENIOR_RATE;
        cout << "\nThe total charges are $" << charges << endl;
    }
    else if (choice != 4)
    {   cout << "\nValid choices are 1 - 4" << endl;
        cout << "Terminating program." << endl;
    }

    return 0;
}
```

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



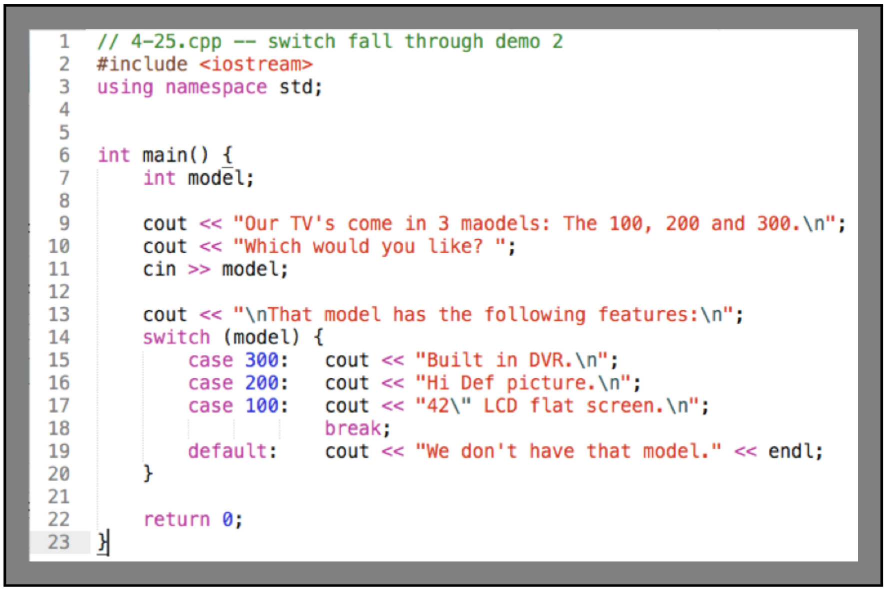
```cpp
// 4-23.cpp -- switch demo
#include <iostream>
using namespace std;


int main() {
    char choice;

    cout << "Enter A, B, C: ";
    cin >> choice;

    switch (choice) {
        case 'A':   cout << "You entered A.\n";
                    break;
        case 'B':   cout << "You entered B.\n";
                    break;
        case 'C':   cout << "You entered A.\n";
                    break;
        default:    cout << "Incorrect entry!\n";
    }

    return 0;
}
```

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



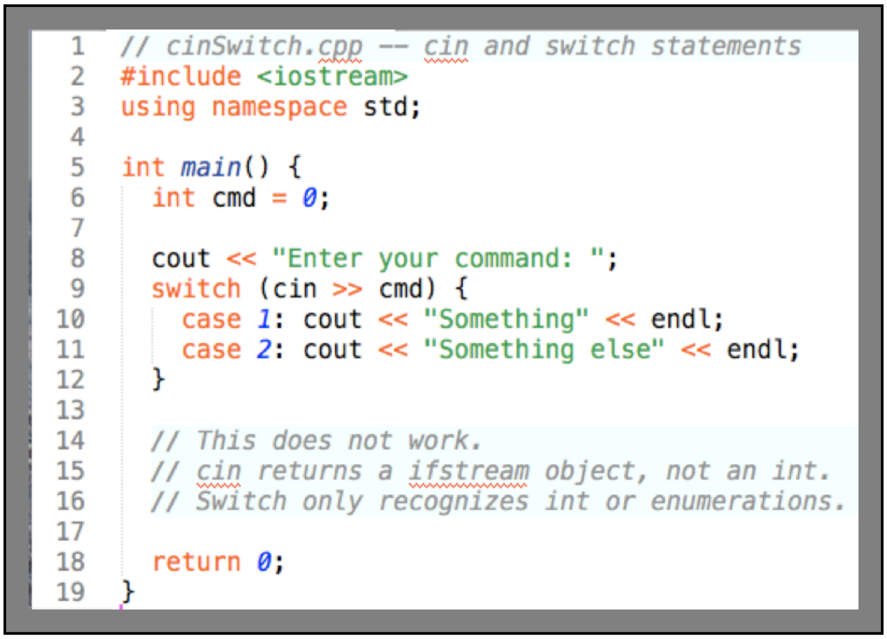
```cpp
// 4-25.cpp -- switch fall through demo
#include <iostream>
using namespace std;


int main() {
    char choice;

    cout << "Enter A, B, C: ";
    cin >> choice;

    switch (choice) {
        case 'a':
        case 'A':   cout << "You entered A.\n";
                    break;
        case 'b':
        case 'B':   cout << "You entered B.\n";
                    break;
        case 'c':
        case 'C':   cout << "You entered C.\n";
                    break;
        default:    cout << "Incorrect entry!\n";
    }

    return 0;
}
```

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



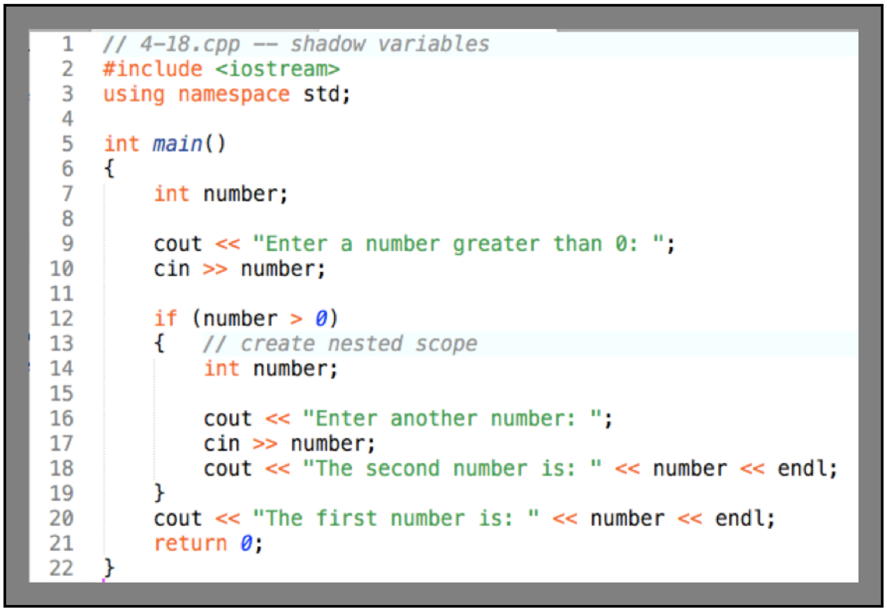
```
// 4-25.cpp -- switch fall through demo 2
#include <iostream>
using namespace std;


int main() {
    int model;

    cout << "Our TV's come in 3 maodels: The 100, 200 and 300.\n";
    cout << "Which would you like? ";
    cin >> model;

    cout << "\nThat model has the following features:\n";
    switch (model) {
        case 300:   cout << "Built in DVR.\n";
        case 200:   cout << "Hi Def picture.\n";
        case 100:   cout << "42\" LCD flat screen.\n";
                    break;
        default:    cout << "We don't have that model." << endl;
    }

    return 0;
}
```

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



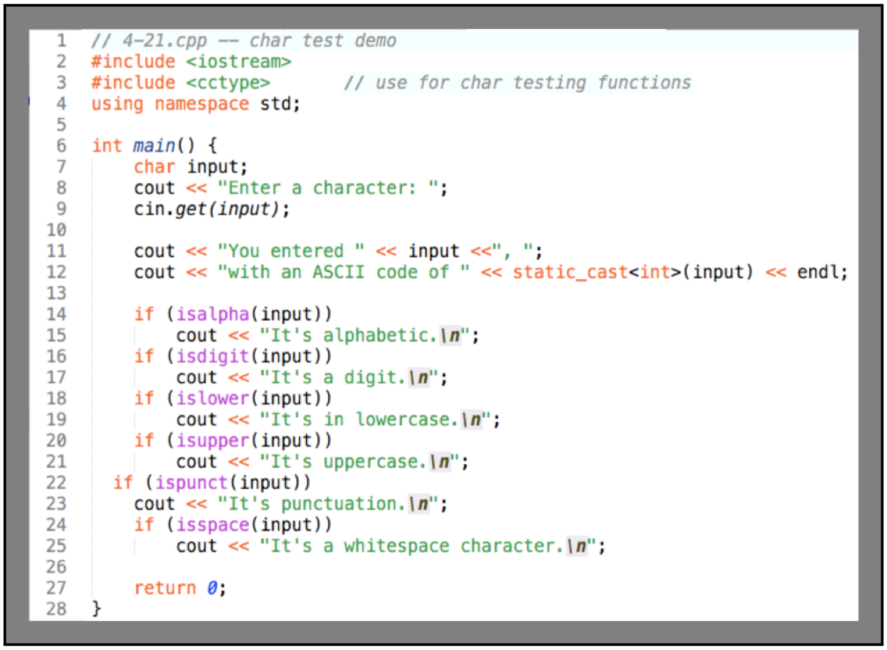
```
// cinSwitch.cpp -- cin and switch statements
#include <iostream>
using namespace std;

int main() {
  int cmd = 0;

  cout << "Enter your command: ";
  switch (cin >> cmd) {
    case 1: cout << "Something" << endl;
    case 2: cout << "Something else" << endl;
  }

  // This does not work.
  // cin returns a ifstream object, not an int.
  // Switch only recognizes int or enumerations.

  return 0;
}
```

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



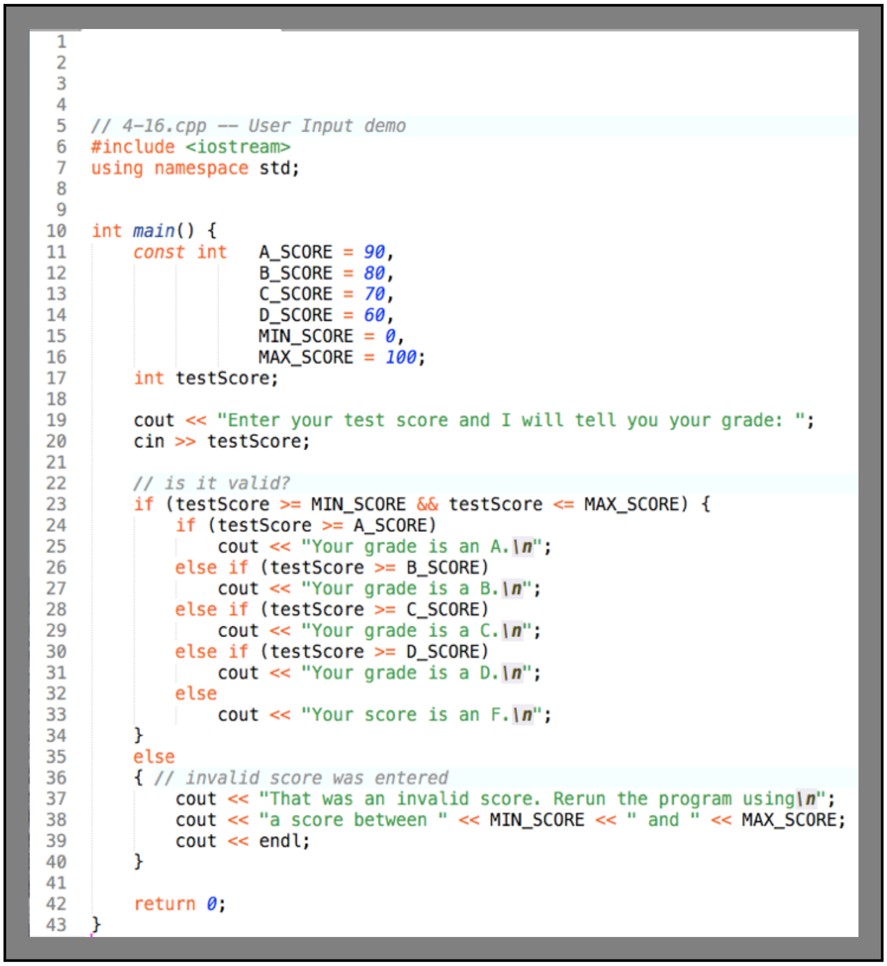
```cpp
// 4-18.cpp -- shadow variables
#include <iostream>
using namespace std;

int main()
{
    int number;

    cout << "Enter a number greater than 0: ";
    cin >> number;

    if (number > 0)
    {   // create nested scope
        int number;

        cout << "Enter another number: ";
        cin >> number;
        cout << "The second number is: " << number << endl;
    }
    cout << "The first number is: " << number << endl;
    return 0;
}
```

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



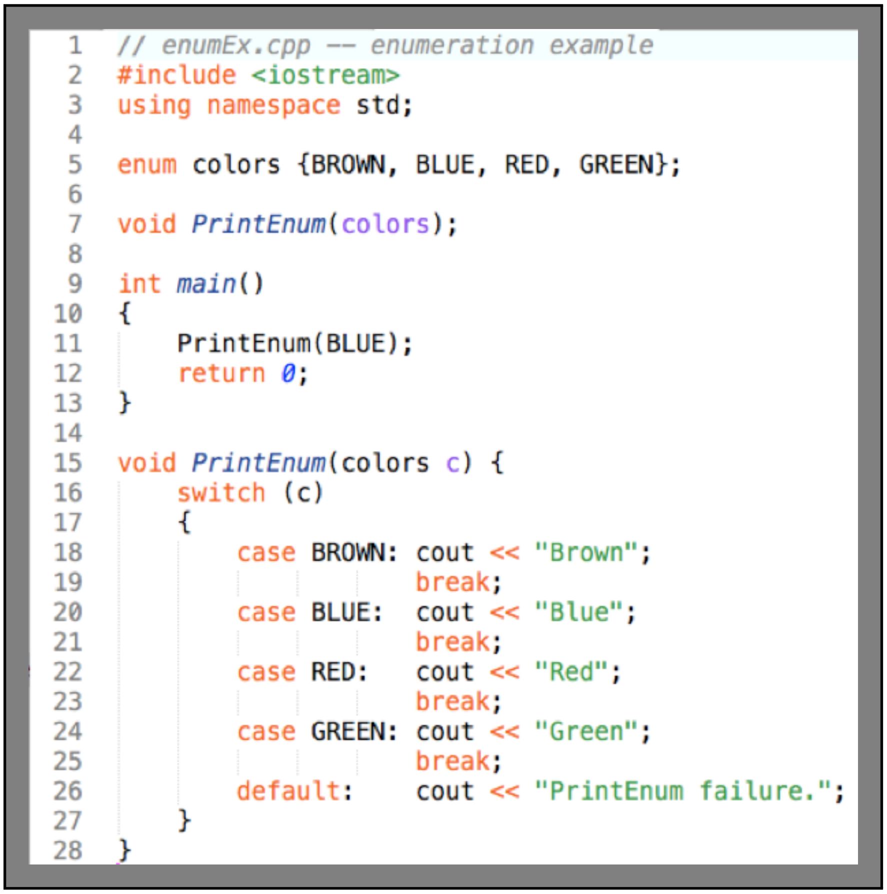
```cpp
// 4-21.cpp -- char test demo
#include <iostream>
#include <cctype>       // use for char testing functions
using namespace std;

int main() {
    char input;
    cout << "Enter a character: ";
    cin.get(input);

    cout << "You entered " << input <<", ";
    cout << "with an ASCII code of " << static_cast<int>(input) << endl;

    if (isalpha(input))
        cout << "It's alphabetic.\n";
    if (isdigit(input))
        cout << "It's a digit.\n";
    if (islower(input))
        cout << "It's in lowercase.\n";
    if (isupper(input))
        cout << "It's uppercase.\n";
  if (ispunct(input))
    cout << "It's punctuation.\n";
    if (isspace(input))
        cout << "It's a whitespace character.\n";

    return 0;
}
```

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



```cpp
// 4-16.cpp -- User Input demo
#include <iostream>
using namespace std;


int main() {
    const int   A_SCORE = 90,
                B_SCORE = 80,
                C_SCORE = 70,
                D_SCORE = 60,
                MIN_SCORE = 0,
                MAX_SCORE = 100;
    int testScore;

    cout << "Enter your test score and I will tell you your grade: ";
    cin >> testScore;

    // is it valid?
    if (testScore >= MIN_SCORE && testScore <= MAX_SCORE) {
        if (testScore >= A_SCORE)
            cout << "Your grade is an A.\n";
        else if (testScore >= B_SCORE)
            cout << "Your grade is a B.\n";
        else if (testScore >= C_SCORE)
            cout << "Your grade is a C.\n";
        else if (testScore >= D_SCORE)
            cout << "Your grade is a D.\n";
        else
            cout << "Your score is an F.\n";
    }
    else
    { // invalid score was entered
        cout << "That was an invalid score. Rerun the program using\n";
        cout << "a score between " << MIN_SCORE << " and " << MAX_SCORE;
        cout << endl;
    }

    return 0;
}
```

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



```
// enumEx.cpp -- enumeration example
#include <iostream>
using namespace std;

enum colors {BROWN, BLUE, RED, GREEN};

void PrintEnum(colors);

int main()
{
    PrintEnum(BLUE);
    return 0;
}

void PrintEnum(colors c) {
    switch (c)
    {
        case BROWN: cout << "Brown";
                    break;
        case BLUE:  cout << "Blue";
                    break;
        case RED:   cout << "Red";
                    break;
        case GREEN: cout << "Green";
                    break;
        default:    cout << "PrintEnum failure.";
    }
}
```

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