

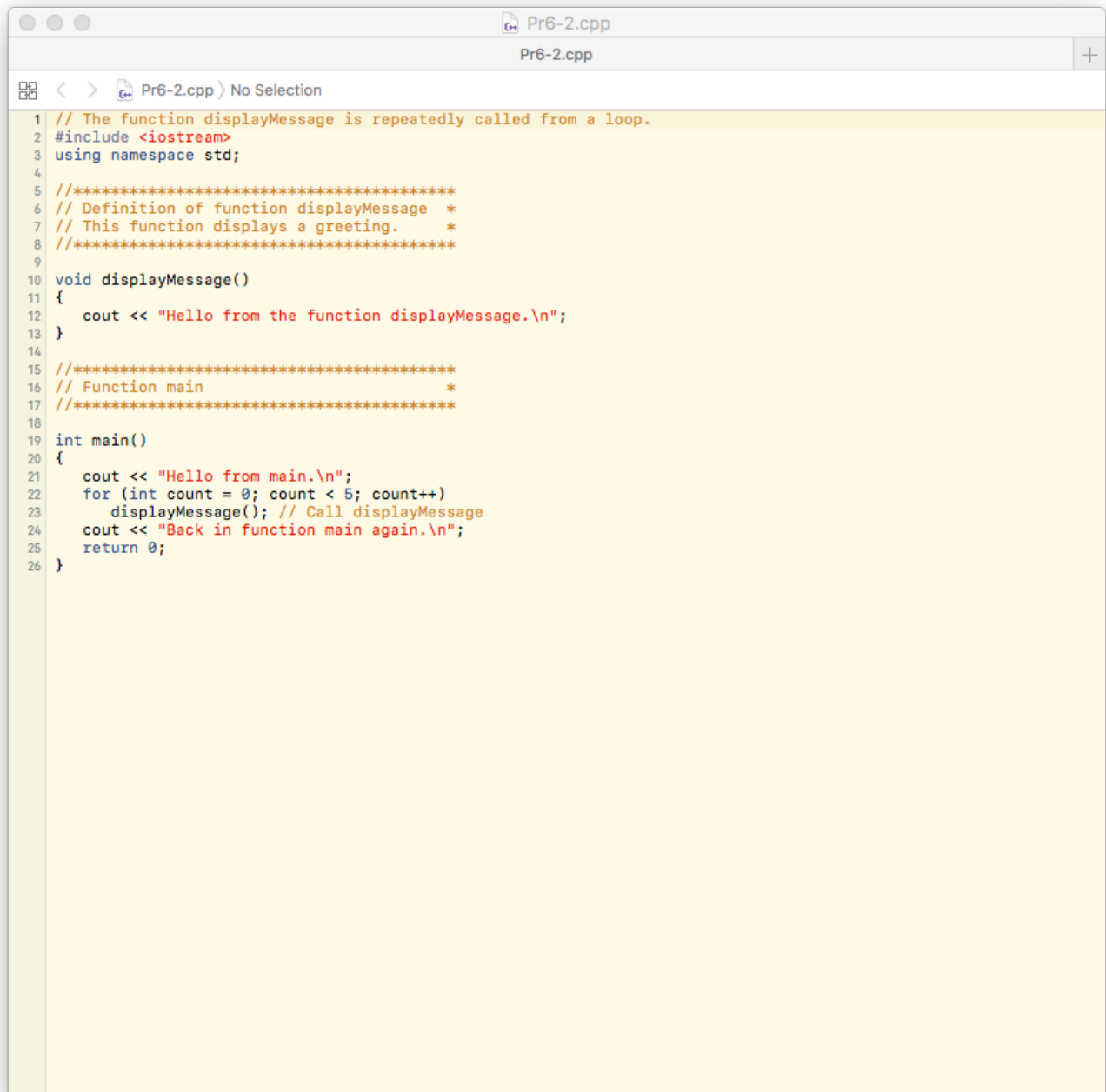
Pr6-1.cpp

Pr6-1.cpp

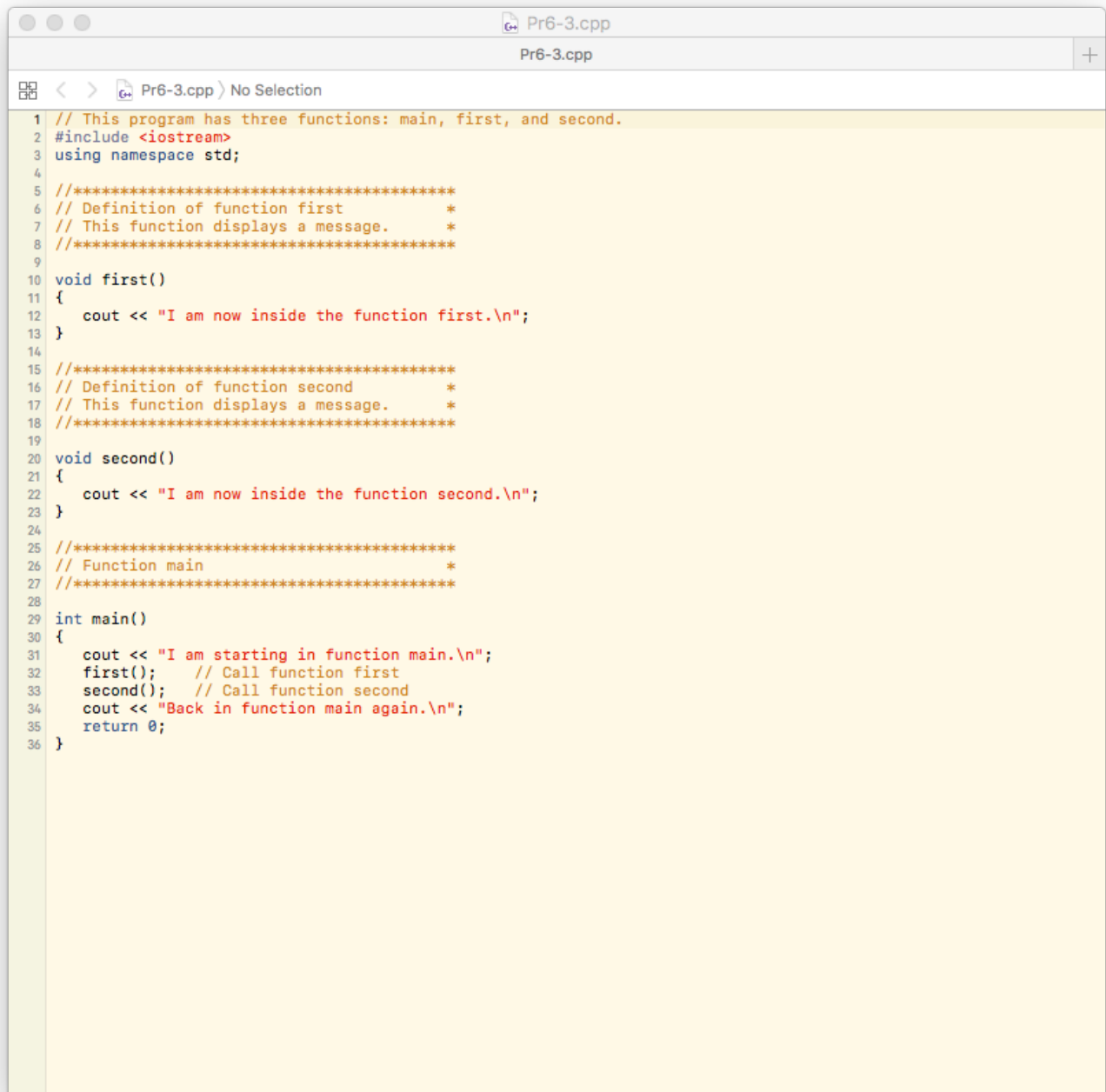


Pr6-1.cpp > No Selection

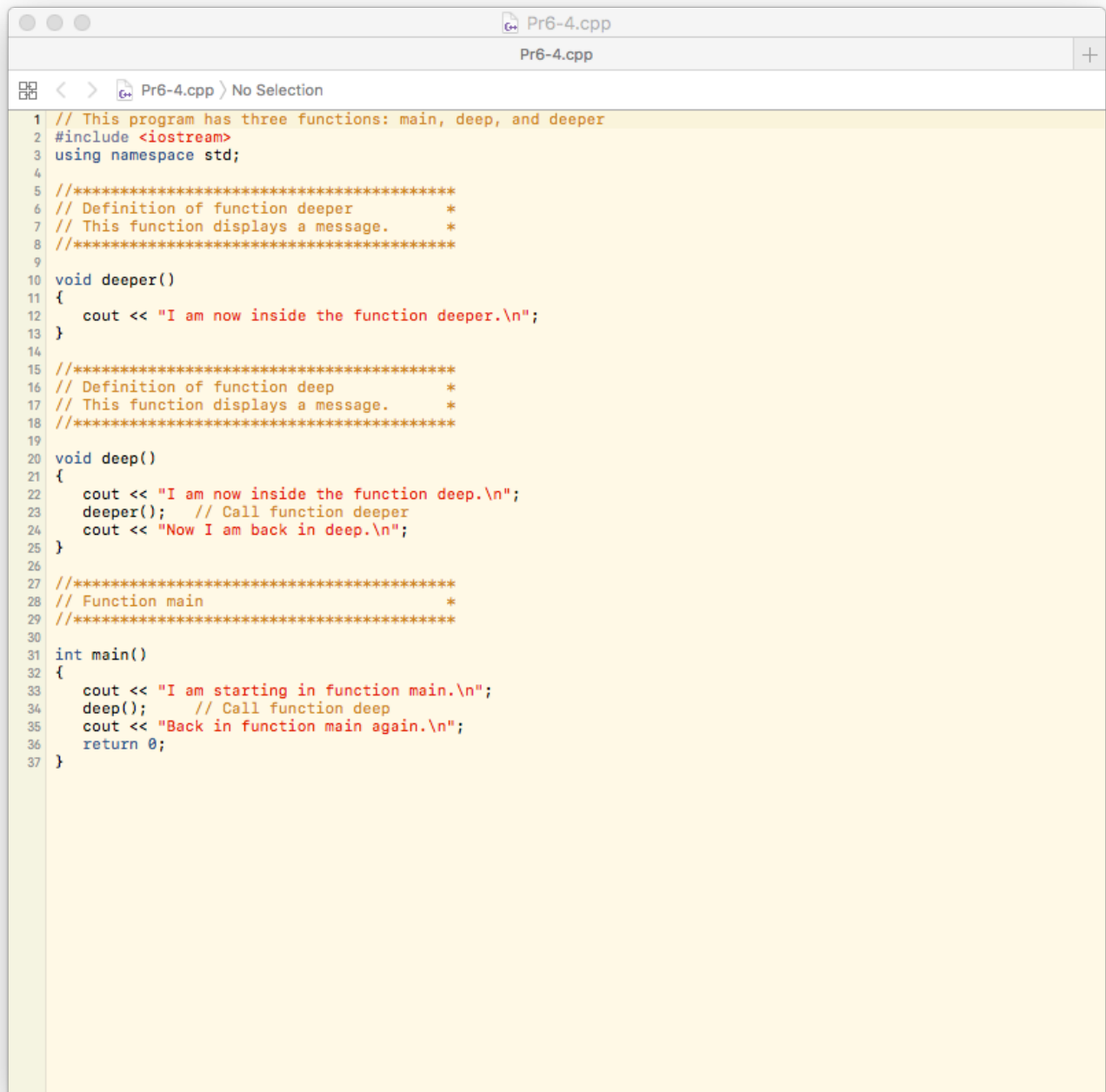
```
1 // This program has two functions: main and displayMessage
2 #include <iostream>
3 using namespace std;
4
5 //*****
6 // Definition of function displayMessage *
7 // This function displays a greeting. *
8 //*****
9
10 void displayMessage()
11 {
12     cout << "Hello from the function displayMessage.\n";
13 }
14
15 //*****
16 // Function main *
17 //*****
18
19 int main()
20 {
21     cout << "Hello from main.\n";
22     displayMessage();
23     cout << "Back in function main again.\n";
24     return 0;
25 }
```



```
1 // The function displayMessage is repeatedly called from a loop.
2 #include <iostream>
3 using namespace std;
4
5 //*****
6 // Definition of function displayMessage *
7 // This function displays a greeting.    *
8 //*****
9
10 void displayMessage()
11 {
12     cout << "Hello from the function displayMessage.\n";
13 }
14
15 //*****
16 // Function main *
17 //*****
18
19 int main()
20 {
21     cout << "Hello from main.\n";
22     for (int count = 0; count < 5; count++)
23         displayMessage(); // Call displayMessage
24     cout << "Back in function main again.\n";
25     return 0;
26 }
```

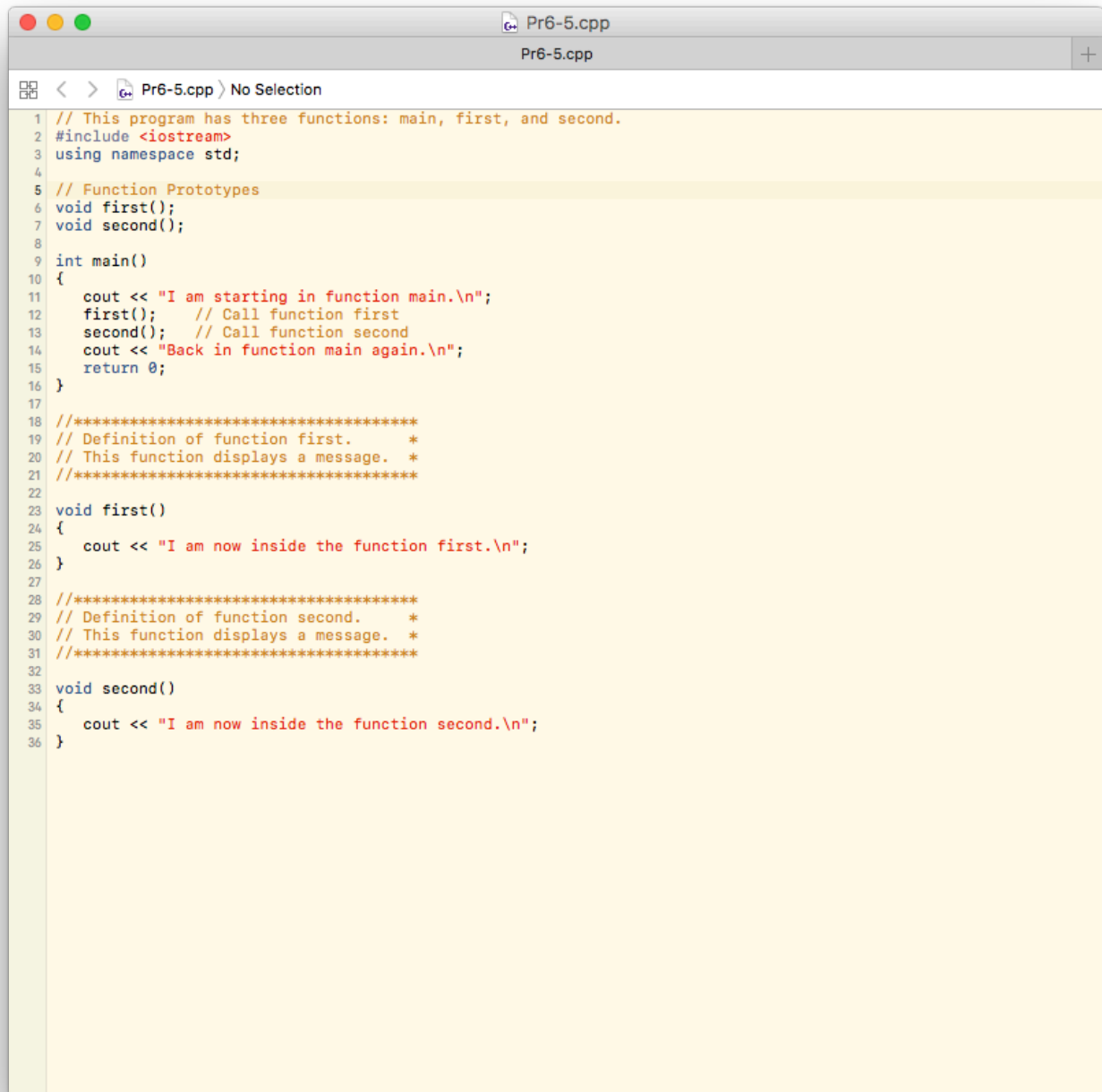


```
1 // This program has three functions: main, first, and second.
2 #include <iostream>
3 using namespace std;
4
5 //*****
6 // Definition of function first      *
7 // This function displays a message. *
8 //*****
9
10 void first()
11 {
12     cout << "I am now inside the function first.\n";
13 }
14
15 //*****
16 // Definition of function second    *
17 // This function displays a message. *
18 //*****
19
20 void second()
21 {
22     cout << "I am now inside the function second.\n";
23 }
24
25 //*****
26 // Function main                    *
27 //*****
28
29 int main()
30 {
31     cout << "I am starting in function main.\n";
32     first();    // Call function first
33     second();   // Call function second
34     cout << "Back in function main again.\n";
35     return 0;
36 }
```

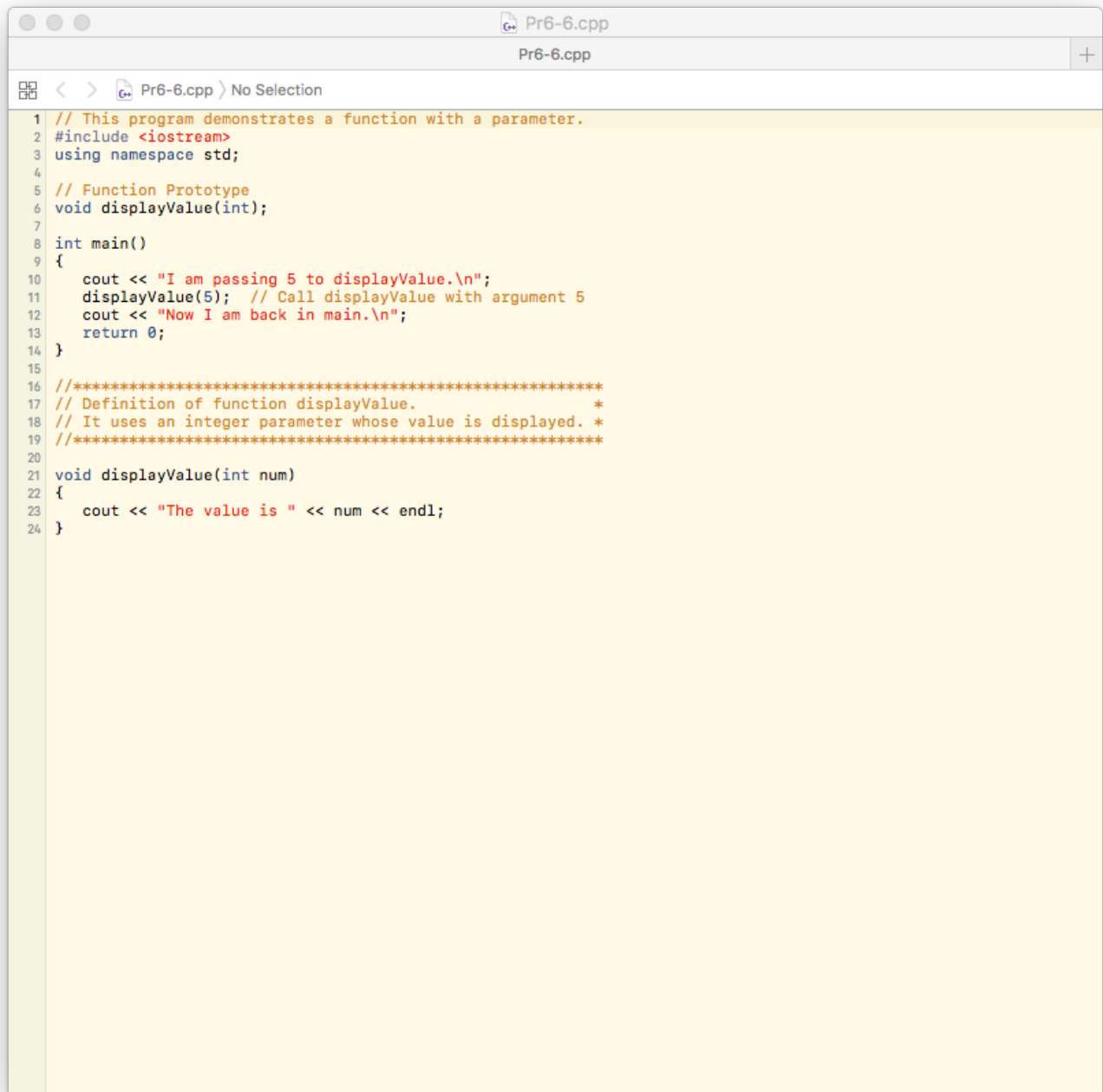


The image shows a code editor window titled "Pr6-4.cpp". The editor contains C++ code for a recursive program. The code defines three functions: `main`, `deep`, and `deeper`. The `main` function calls `deep`, which calls `deeper`, and `deeper` calls back to `deep`. The code uses `cout` to display messages at each step.

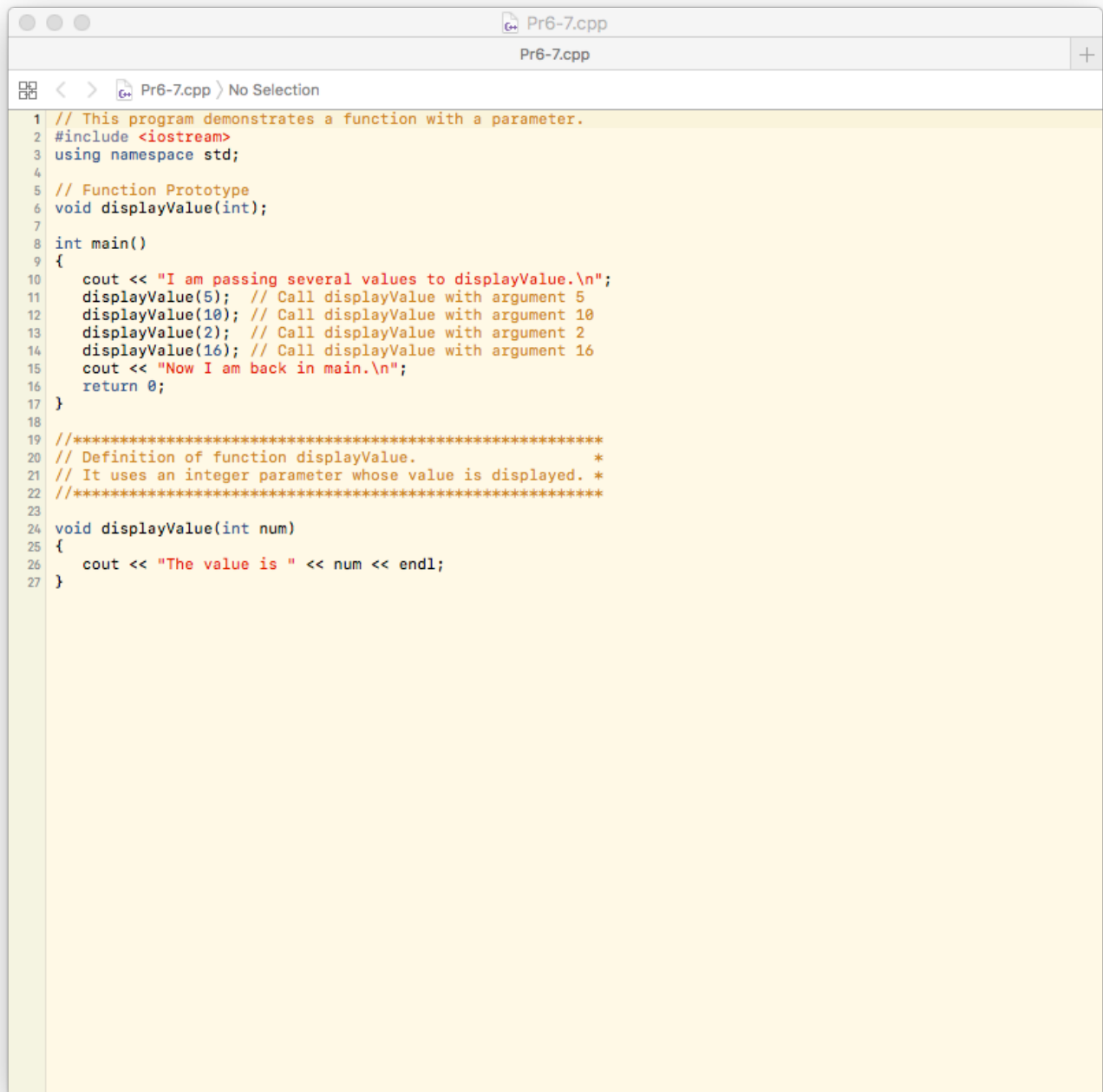
```
1 // This program has three functions: main, deep, and deeper
2 #include <iostream>
3 using namespace std;
4
5 //*****
6 // Definition of function deeper          *
7 // This function displays a message.      *
8 //*****
9
10 void deeper()
11 {
12     cout << "I am now inside the function deeper.\n";
13 }
14
15 //*****
16 // Definition of function deep           *
17 // This function displays a message.      *
18 //*****
19
20 void deep()
21 {
22     cout << "I am now inside the function deep.\n";
23     deeper(); // Call function deeper
24     cout << "Now I am back in deep.\n";
25 }
26
27 //*****
28 // Function main                         *
29 //*****
30
31 int main()
32 {
33     cout << "I am starting in function main.\n";
34     deep(); // Call function deep
35     cout << "Back in function main again.\n";
36     return 0;
37 }
```



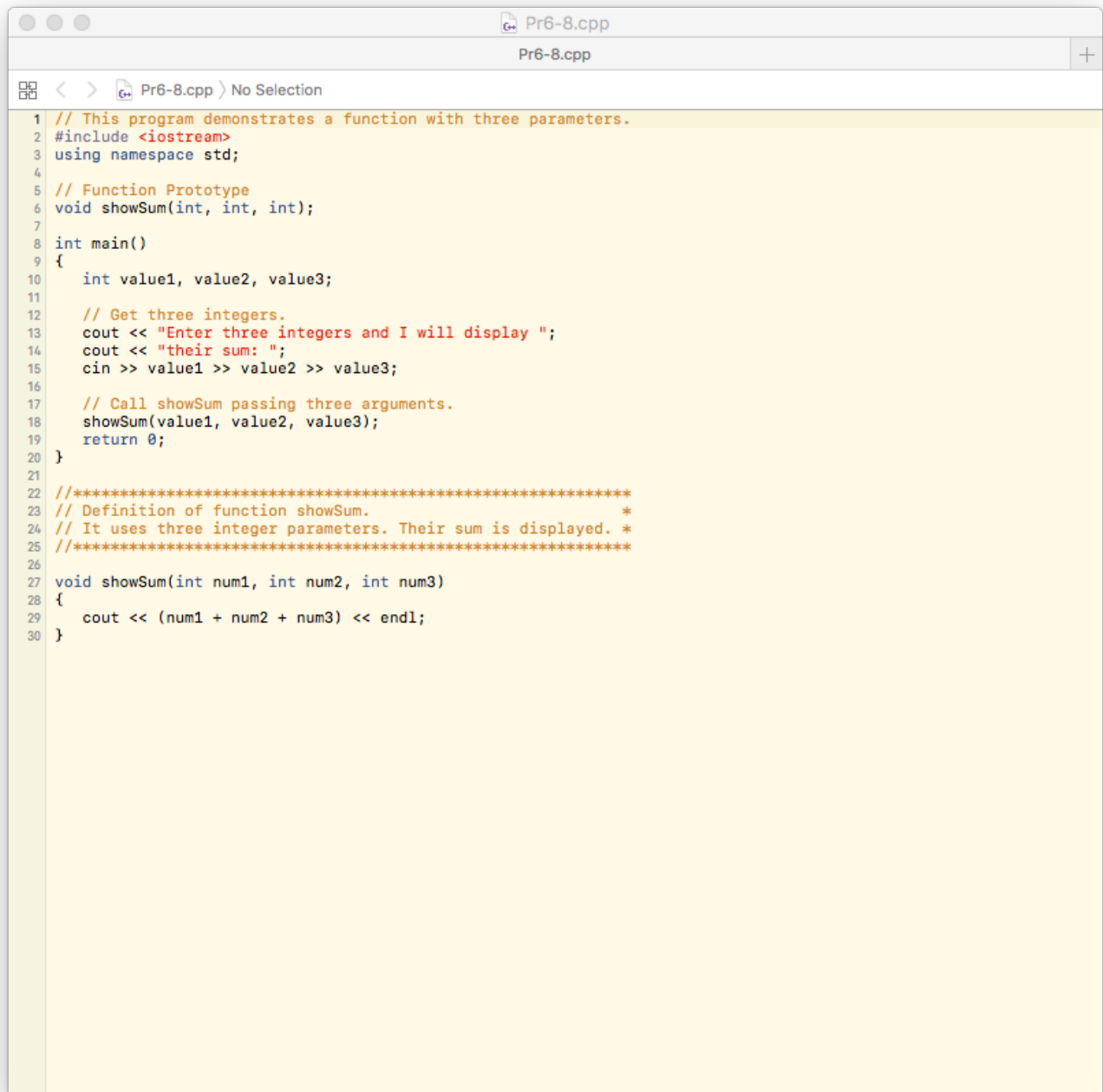
```
1 // This program has three functions: main, first, and second.
2 #include <iostream>
3 using namespace std;
4
5 // Function Prototypes
6 void first();
7 void second();
8
9 int main()
10 {
11     cout << "I am starting in function main.\n";
12     first(); // Call function first
13     second(); // Call function second
14     cout << "Back in function main again.\n";
15     return 0;
16 }
17
18 //*****
19 // Definition of function first.      *
20 // This function displays a message. *
21 //*****
22
23 void first()
24 {
25     cout << "I am now inside the function first.\n";
26 }
27
28 //*****
29 // Definition of function second.    *
30 // This function displays a message. *
31 //*****
32
33 void second()
34 {
35     cout << "I am now inside the function second.\n";
36 }
```



```
1 // This program demonstrates a function with a parameter.
2 #include <iostream>
3 using namespace std;
4
5 // Function Prototype
6 void displayValue(int);
7
8 int main()
9 {
10     cout << "I am passing 5 to displayValue.\n";
11     displayValue(5); // Call displayValue with argument 5
12     cout << "Now I am back in main.\n";
13     return 0;
14 }
15
16 //*****
17 // Definition of function displayValue. *
18 // It uses an integer parameter whose value is displayed. *
19 //*****
20
21 void displayValue(int num)
22 {
23     cout << "The value is " << num << endl;
24 }
```

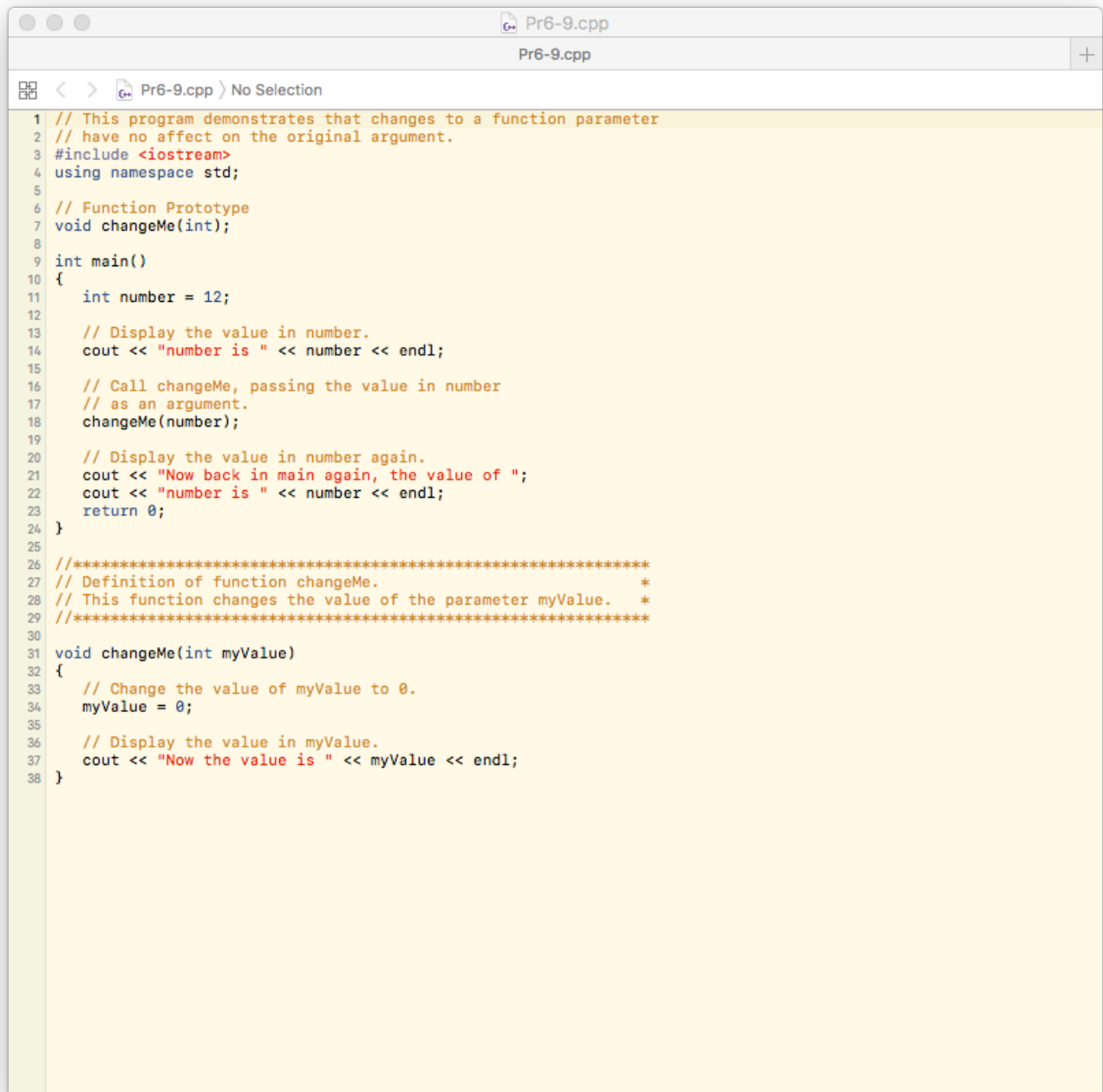


```
1 // This program demonstrates a function with a parameter.
2 #include <iostream>
3 using namespace std;
4
5 // Function Prototype
6 void displayValue(int);
7
8 int main()
9 {
10     cout << "I am passing several values to displayValue.\n";
11     displayValue(5); // Call displayValue with argument 5
12     displayValue(10); // Call displayValue with argument 10
13     displayValue(2); // Call displayValue with argument 2
14     displayValue(16); // Call displayValue with argument 16
15     cout << "Now I am back in main.\n";
16     return 0;
17 }
18
19 //*****
20 // Definition of function displayValue.
21 // It uses an integer parameter whose value is displayed. *
22 //*****
23
24 void displayValue(int num)
25 {
26     cout << "The value is " << num << endl;
27 }
```

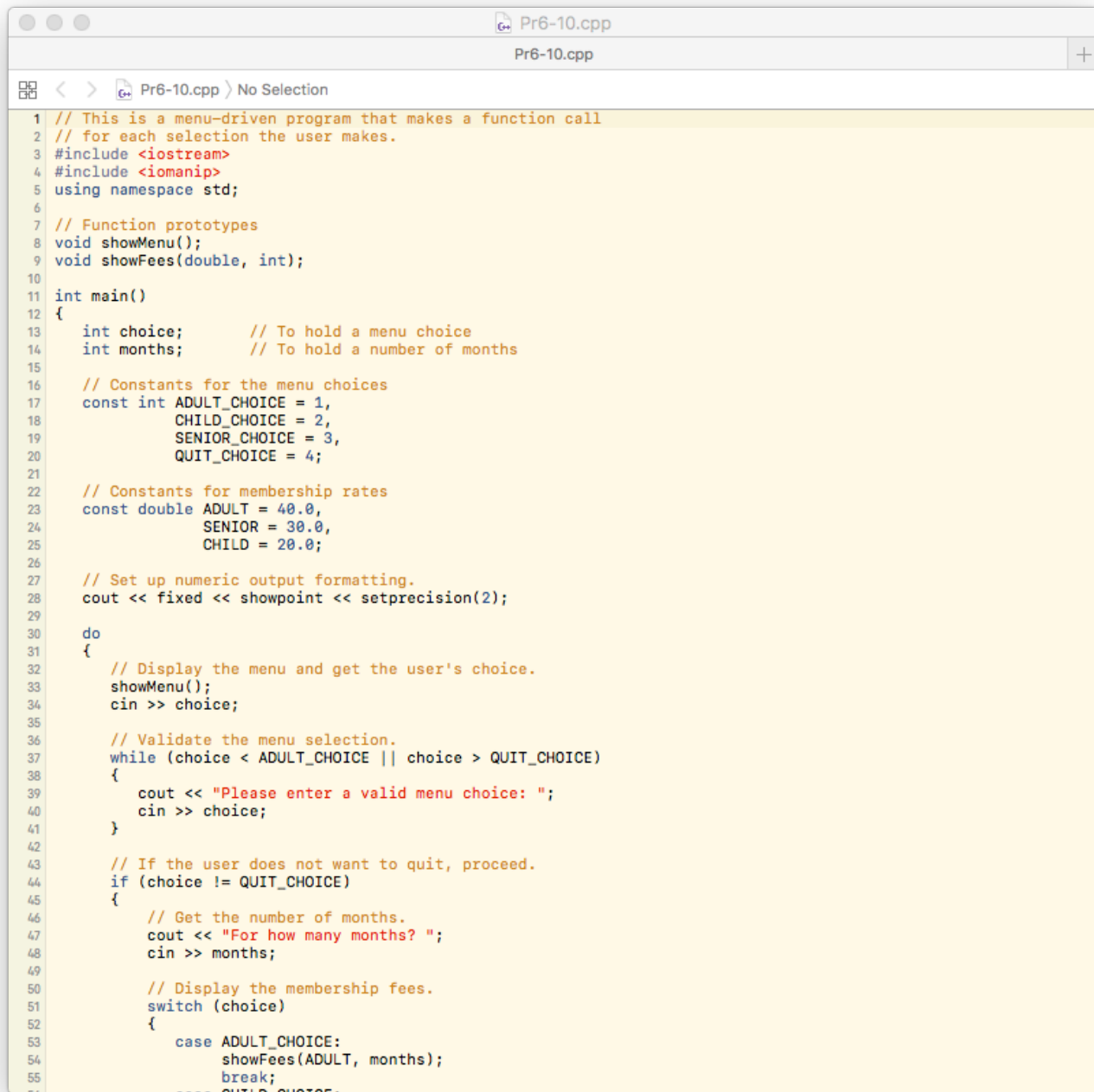


The image shows a screenshot of a C++ code editor window. The window has a title bar with three standard macOS window control buttons (red, yellow, green) on the left and the filename "Pr6-8.cpp" in the center. Below the title bar is a toolbar with icons for file operations and a dropdown menu showing "Pr6-8.cpp" and "No Selection". The main area of the window contains C++ code with line numbers 1 through 30 on the left margin. The code defines a function `showSum` and uses it in the `main` function. The code is as follows:

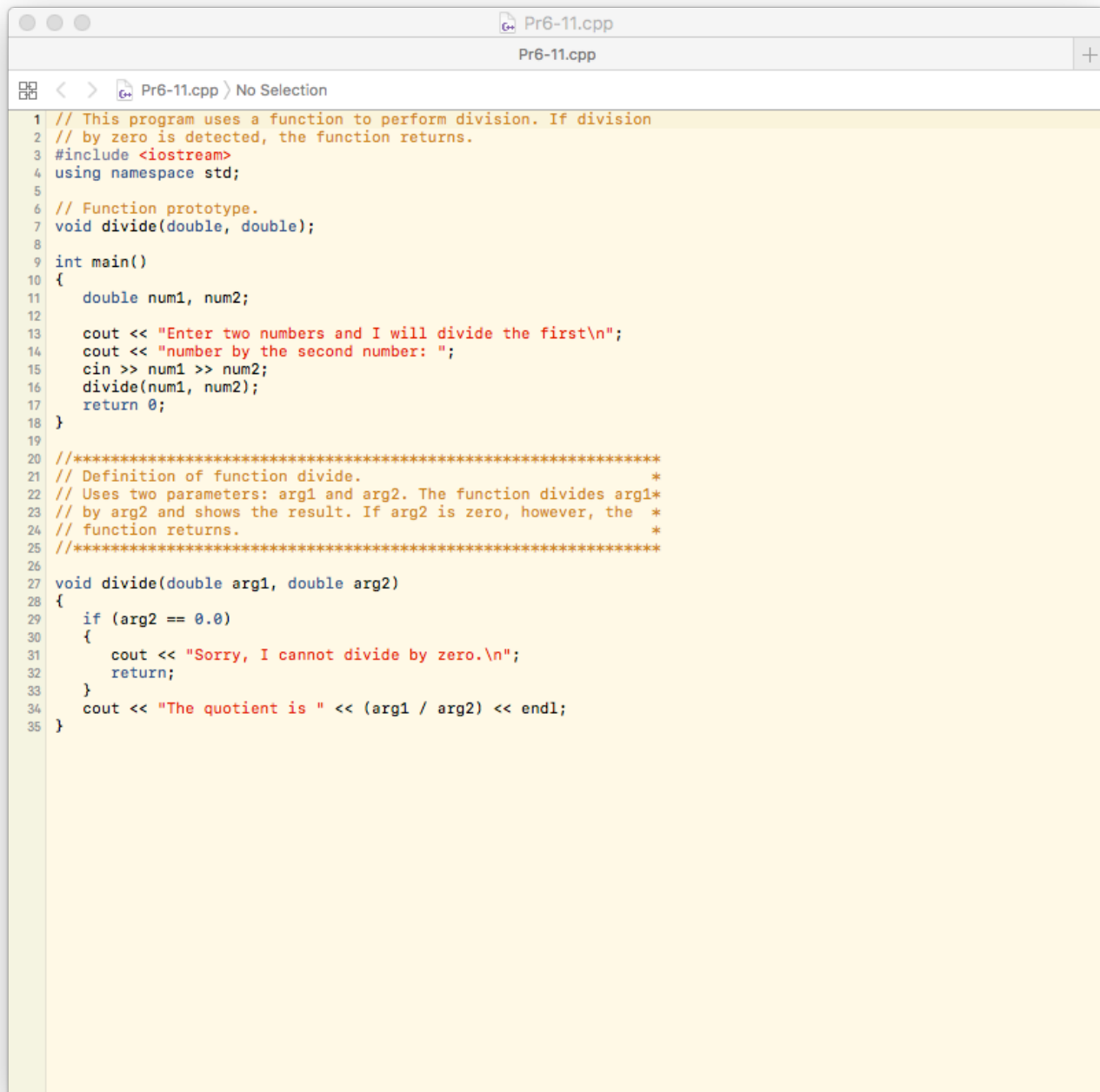
```
1 // This program demonstrates a function with three parameters.
2 #include <iostream>
3 using namespace std;
4
5 // Function Prototype
6 void showSum(int, int, int);
7
8 int main()
9 {
10     int value1, value2, value3;
11
12     // Get three integers.
13     cout << "Enter three integers and I will display ";
14     cout << "their sum: ";
15     cin >> value1 >> value2 >> value3;
16
17     // Call showSum passing three arguments.
18     showSum(value1, value2, value3);
19     return 0;
20 }
21
22 //*****
23 // Definition of function showSum.
24 // It uses three integer parameters. Their sum is displayed. *
25 //*****
26
27 void showSum(int num1, int num2, int num3)
28 {
29     cout << (num1 + num2 + num3) << endl;
30 }
```

```
1 // This program demonstrates that changes to a function parameter
2 // have no affect on the original argument.
3 #include <iostream>
4 using namespace std;
5
6 // Function Prototype
7 void changeMe(int);
8
9 int main()
10 {
11     int number = 12;
12
13     // Display the value in number.
14     cout << "number is " << number << endl;
15
16     // Call changeMe, passing the value in number
17     // as an argument.
18     changeMe(number);
19
20     // Display the value in number again.
21     cout << "Now back in main again, the value of ";
22     cout << "number is " << number << endl;
23     return 0;
24 }
25
26 //*****
27 // Definition of function changeMe.
28 // This function changes the value of the parameter myValue.
29 //*****
30
31 void changeMe(int myValue)
32 {
33     // Change the value of myValue to 0.
34     myValue = 0;
35
36     // Display the value in myValue.
37     cout << "Now the value is " << myValue << endl;
38 }
```

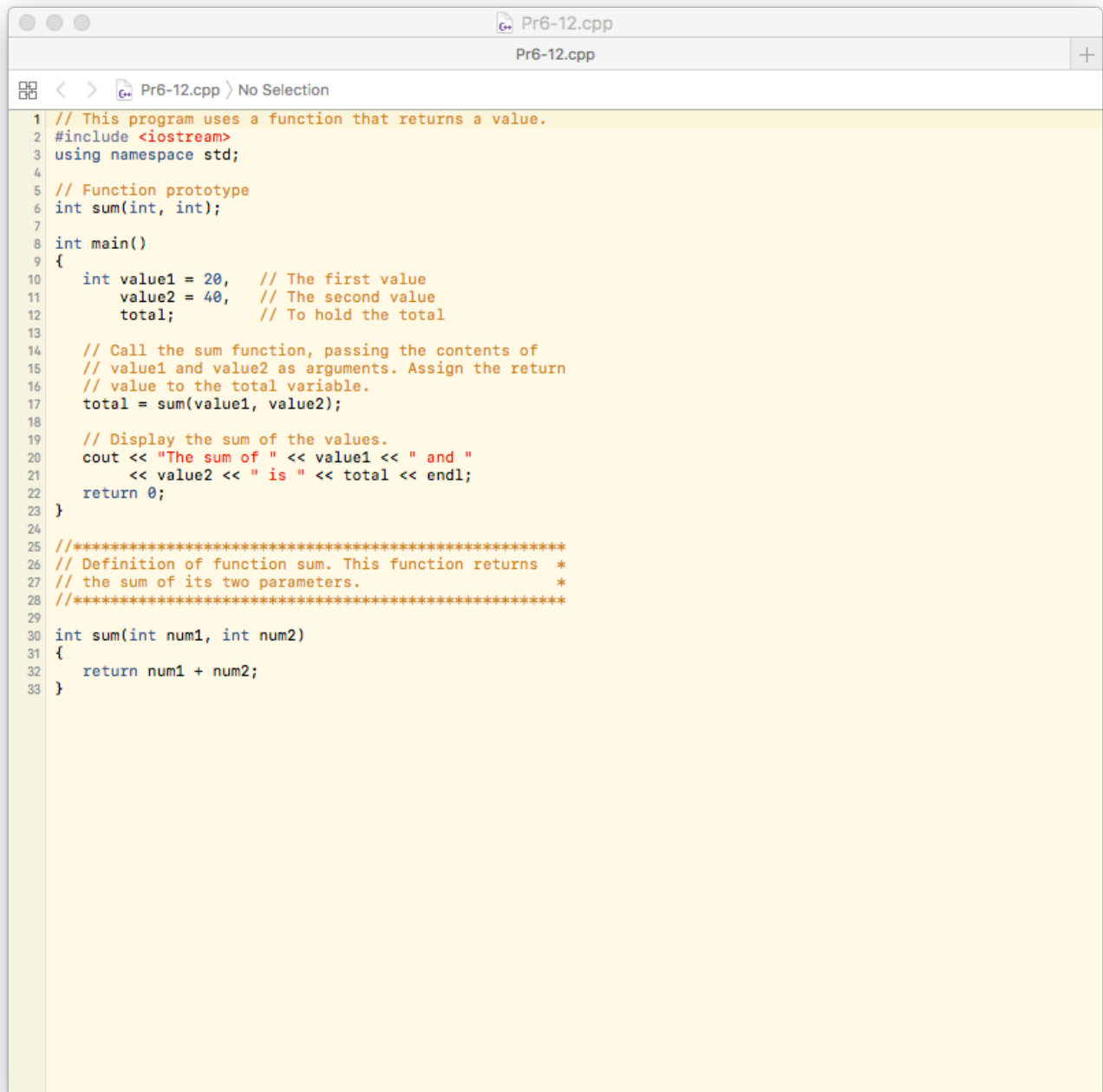


```
1 // This is a menu-driven program that makes a function call
2 // for each selection the user makes.
3 #include <iostream>
4 #include <iomanip>
5 using namespace std;
6
7 // Function prototypes
8 void showMenu();
9 void showFees(double, int);
10
11 int main()
12 {
13     int choice;        // To hold a menu choice
14     int months;        // To hold a number of months
15
16     // Constants for the menu choices
17     const int ADULT_CHOICE = 1,
18             CHILD_CHOICE = 2,
19             SENIOR_CHOICE = 3,
20             QUIT_CHOICE = 4;
21
22     // Constants for membership rates
23     const double ADULT = 40.0,
24                 SENIOR = 30.0,
25                 CHILD = 20.0;
26
27     // Set up numeric output formatting.
28     cout << fixed << showpoint << setprecision(2);
29
30     do
31     {
32         // Display the menu and get the user's choice.
33         showMenu();
34         cin >> choice;
35
36         // Validate the menu selection.
37         while (choice < ADULT_CHOICE || choice > QUIT_CHOICE)
38         {
39             cout << "Please enter a valid menu choice: ";
40             cin >> choice;
41         }
42
43         // If the user does not want to quit, proceed.
44         if (choice != QUIT_CHOICE)
45         {
46             // Get the number of months.
47             cout << "For how many months? ";
48             cin >> months;
49
50             // Display the membership fees.
51             switch (choice)
52             {
53                 case ADULT_CHOICE:
54                     showFees(ADULT, months);
55                     break;
56                 case CHILD_CHOICE:
```



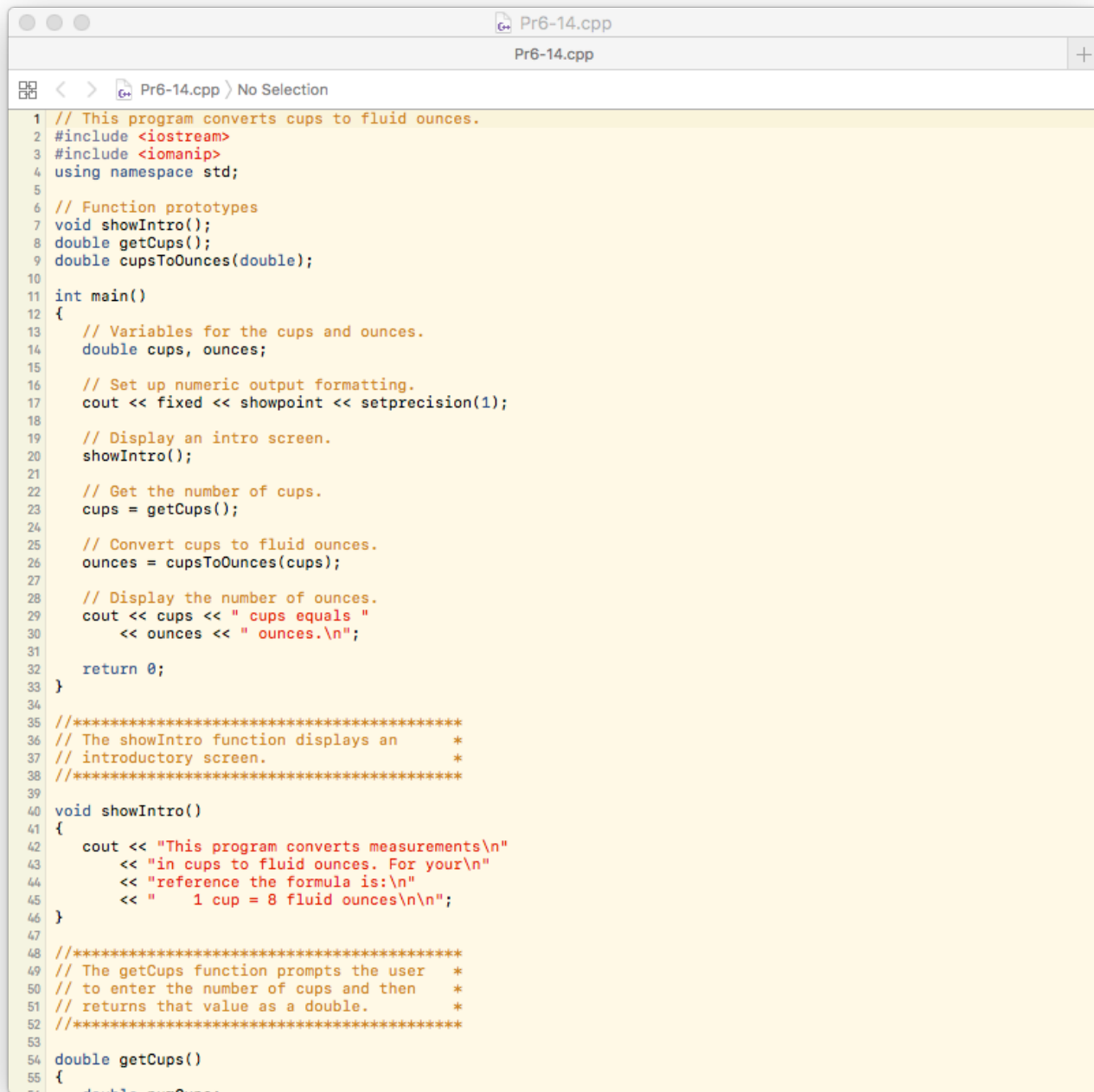
The image shows a C++ IDE window titled "Pr6-11.cpp". The code is a C++ program that uses a function to perform division. It includes the `<iostream>` header and uses the `std` namespace. The `main` function prompts the user to enter two numbers and calls the `divide` function. The `divide` function is defined to take two `double` arguments. It checks if the second argument is zero; if so, it prints an error message and returns. Otherwise, it calculates the quotient and prints it.

```
1 // This program uses a function to perform division. If division
2 // by zero is detected, the function returns.
3 #include <iostream>
4 using namespace std;
5
6 // Function prototype.
7 void divide(double, double);
8
9 int main()
10 {
11     double num1, num2;
12
13     cout << "Enter two numbers and I will divide the first\n";
14     cout << "number by the second number: ";
15     cin >> num1 >> num2;
16     divide(num1, num2);
17     return 0;
18 }
19
20 //*****
21 // Definition of function divide.
22 // Uses two parameters: arg1 and arg2. The function divides arg1*
23 // by arg2 and shows the result. If arg2 is zero, however, the
24 // function returns.
25 //*****
26
27 void divide(double arg1, double arg2)
28 {
29     if (arg2 == 0.0)
30     {
31         cout << "Sorry, I cannot divide by zero.\n";
32         return;
33     }
34     cout << "The quotient is " << (arg1 / arg2) << endl;
35 }
```

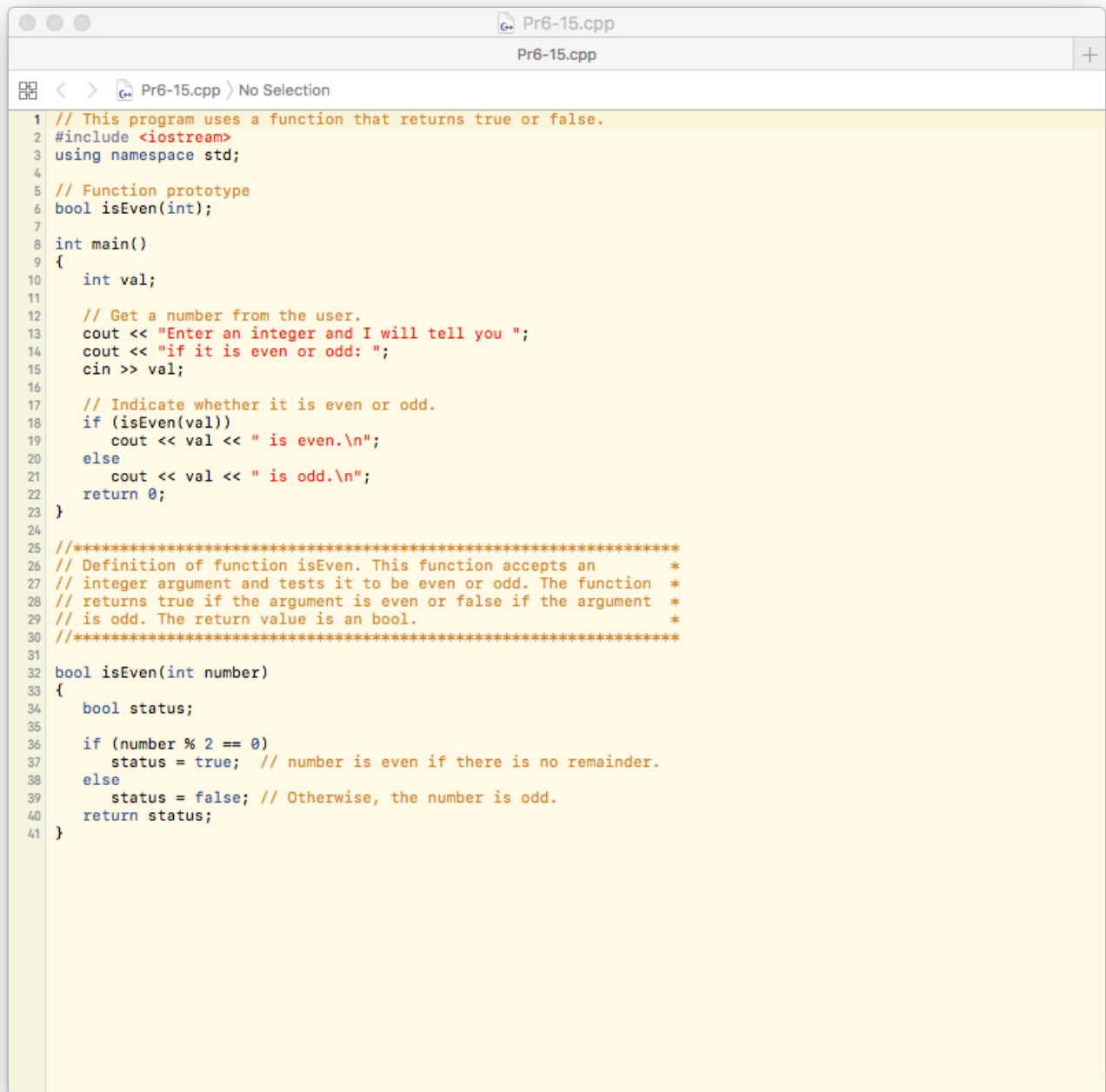


```
1 // This program uses a function that returns a value.
2 #include <iostream>
3 using namespace std;
4
5 // Function prototype
6 int sum(int, int);
7
8 int main()
9 {
10     int value1 = 20,    // The first value
11         value2 = 40,    // The second value
12         total;          // To hold the total
13
14     // Call the sum function, passing the contents of
15     // value1 and value2 as arguments. Assign the return
16     // value to the total variable.
17     total = sum(value1, value2);
18
19     // Display the sum of the values.
20     cout << "The sum of " << value1 << " and "
21          << value2 << " is " << total << endl;
22     return 0;
23 }
24
25 //*****
26 // Definition of function sum. This function returns *
27 // the sum of its two parameters.                  *
28 //*****
29
30 int sum(int num1, int num2)
31 {
32     return num1 + num2;
33 }
```

```
Pr6-13.cpp
Pr6-13.cpp
main()
1 // This program demonstrates two value-returning functions.
2 // The square function is called in a mathematical statement.
3 #include <iostream>
4 #include <iomanip>
5 using namespace std;
6
7 //Function prototypes
8 double getRadius();
9 double square(double);
10
11 int main()
12 {
13     const double PI = 3.14159; // Constant for pi
14     double radius;             // To hold the circle's radius
15     double area;               // To hold the circle's area
16
17     // Set the numeric output formatting.
18     cout << fixed << showpoint << setprecision(2);
19
20     // Get the radius of the circle.
21     cout << "This program calculates the area of ";
22     cout << "a circle.\n";
23     radius = getRadius();
24
25     // Calculate the area of the circle.
26     area = PI * square(radius);
27
28     // Display the area.
29     cout << "The area is " << area << endl;
30     return 0;
31 }
32
33 //*****
34 // Definition of function getRadius.
35 // This function asks the user to enter the radius of
36 // the circle and then returns that number as a double.*
37 //*****
38
39 double getRadius()
40 {
41     double rad;
42
43     cout << "Enter the radius of the circle: ";
44     cin >> rad;
45     return rad;
46 }
47
48 //*****
49 // Definition of function square.
50 // This function accepts a double argument and returns
51 // the square of the argument as a double.
52 //*****
53
54 double square(double number)
55 {
56     return number * number;
```

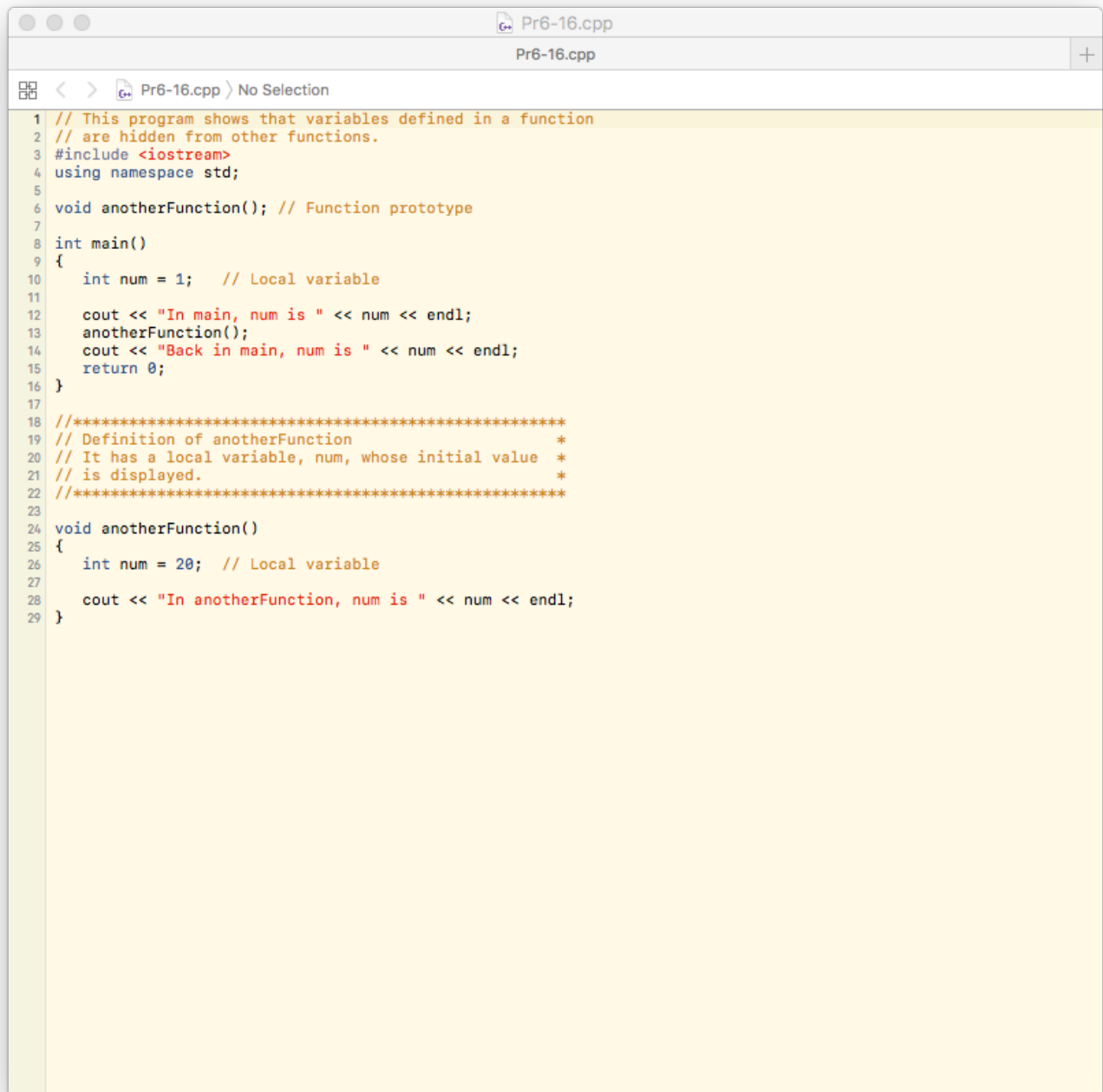


```
1 // This program converts cups to fluid ounces.
2 #include <iostream>
3 #include <iomanip>
4 using namespace std;
5
6 // Function prototypes
7 void showIntro();
8 double getCups();
9 double cupsToOunces(double);
10
11 int main()
12 {
13     // Variables for the cups and ounces.
14     double cups, ounces;
15
16     // Set up numeric output formatting.
17     cout << fixed << showpoint << setprecision(1);
18
19     // Display an intro screen.
20     showIntro();
21
22     // Get the number of cups.
23     cups = getCups();
24
25     // Convert cups to fluid ounces.
26     ounces = cupsToOunces(cups);
27
28     // Display the number of ounces.
29     cout << cups << " cups equals "
30          << ounces << " ounces.\n";
31
32     return 0;
33 }
34
35 //*****
36 // The showIntro function displays an      *
37 // introductory screen.                    *
38 //*****
39
40 void showIntro()
41 {
42     cout << "This program converts measurements\n"
43          << "in cups to fluid ounces. For your\n"
44          << "reference the formula is:\n"
45          << "    1 cup = 8 fluid ounces\n\n";
46 }
47
48 //*****
49 // The getCups function prompts the user  *
50 // to enter the number of cups and then  *
51 // returns that value as a double.        *
52 //*****
53
54 double getCups()
55 {
56     double numCups;
```



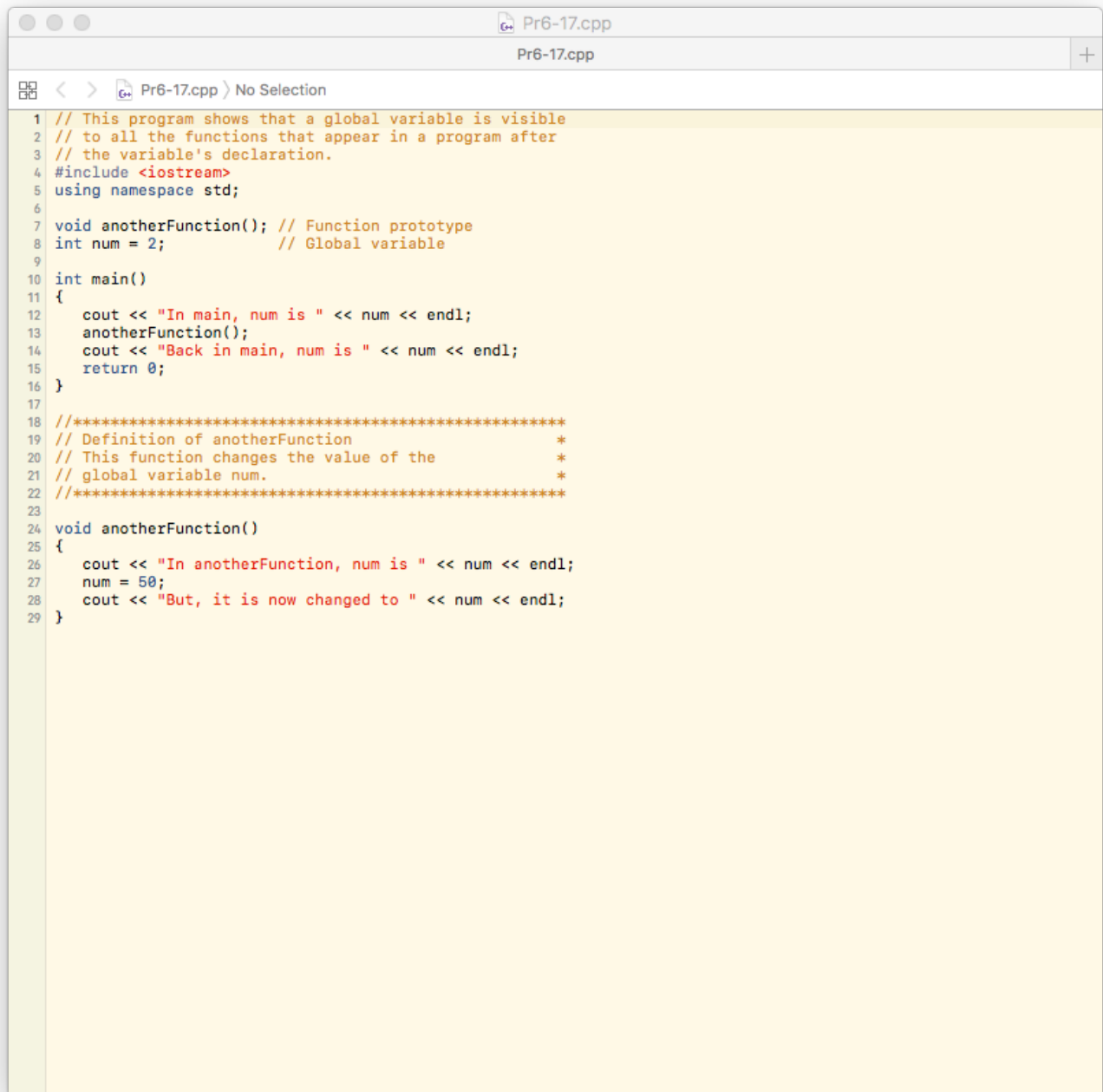
The image shows a C++ IDE window titled "Pr6-15.cpp". The code defines a function `isEven` that takes an integer and returns a boolean. The `main` function prompts the user to enter an integer and then checks if it is even or odd using the `isEven` function. The code is as follows:

```
1 // This program uses a function that returns true or false.
2 #include <iostream>
3 using namespace std;
4
5 // Function prototype
6 bool isEven(int);
7
8 int main()
9 {
10     int val;
11
12     // Get a number from the user.
13     cout << "Enter an integer and I will tell you ";
14     cout << "if it is even or odd: ";
15     cin >> val;
16
17     // Indicate whether it is even or odd.
18     if (isEven(val))
19         cout << val << " is even.\n";
20     else
21         cout << val << " is odd.\n";
22     return 0;
23 }
24
25 //*****
26 // Definition of function isEven. This function accepts an *
27 // integer argument and tests it to be even or odd. The function *
28 // returns true if the argument is even or false if the argument *
29 // is odd. The return value is an bool. *
30 //*****
31
32 bool isEven(int number)
33 {
34     bool status;
35
36     if (number % 2 == 0)
37         status = true; // number is even if there is no remainder.
38     else
39         status = false; // Otherwise, the number is odd.
40     return status;
41 }
```

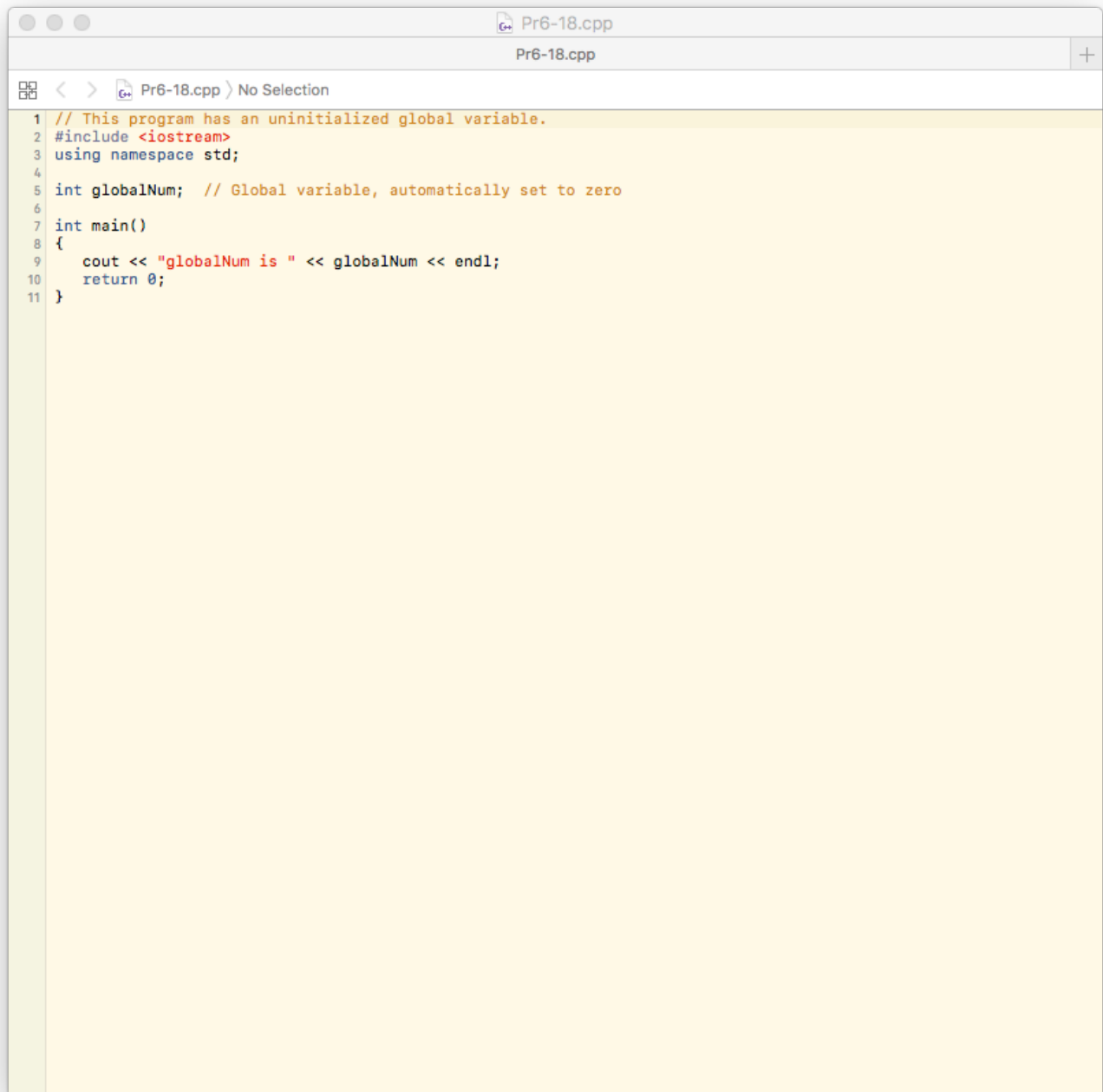


The image shows a screenshot of a C++ IDE window titled "Pr6-16.cpp". The window has a standard macOS-style title bar with three colored buttons (red, yellow, green) on the left. Below the title bar is a toolbar with icons for file operations and a search icon. The main area of the window displays the source code of a C++ program. The code is as follows:

```
1 // This program shows that variables defined in a function
2 // are hidden from other functions.
3 #include <iostream>
4 using namespace std;
5
6 void anotherFunction(); // Function prototype
7
8 int main()
9 {
10     int num = 1;    // Local variable
11
12     cout << "In main, num is " << num << endl;
13     anotherFunction();
14     cout << "Back in main, num is " << num << endl;
15     return 0;
16 }
17
18 //*****
19 // Definition of anotherFunction      *
20 // It has a local variable, num, whose initial value *
21 // is displayed.                        *
22 //*****
23
24 void anotherFunction()
25 {
26     int num = 20;    // Local variable
27
28     cout << "In anotherFunction, num is " << num << endl;
29 }
```

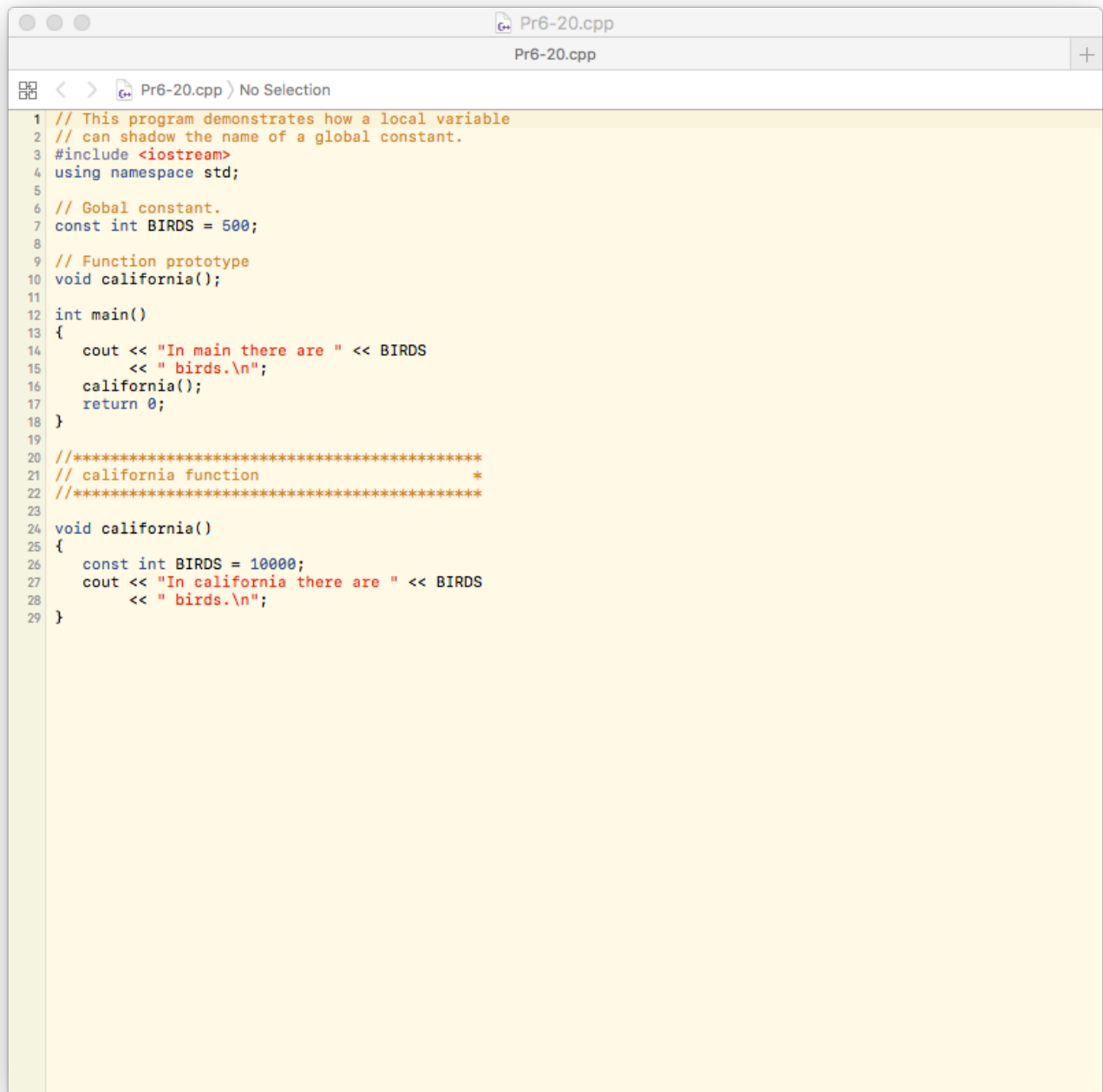
```
1 // This program shows that a global variable is visible
2 // to all the functions that appear in a program after
3 // the variable's declaration.
4 #include <iostream>
5 using namespace std;
6
7 void anotherFunction(); // Function prototype
8 int num = 2;           // Global variable
9
10 int main()
11 {
12     cout << "In main, num is " << num << endl;
13     anotherFunction();
14     cout << "Back in main, num is " << num << endl;
15     return 0;
16 }
17
18 //*****
19 // Definition of anotherFunction          *
20 // This function changes the value of the  *
21 // global variable num.                   *
22 //*****
23
24 void anotherFunction()
25 {
26     cout << "In anotherFunction, num is " << num << endl;
27     num = 50;
28     cout << "But, it is now changed to " << num << endl;
29 }
```



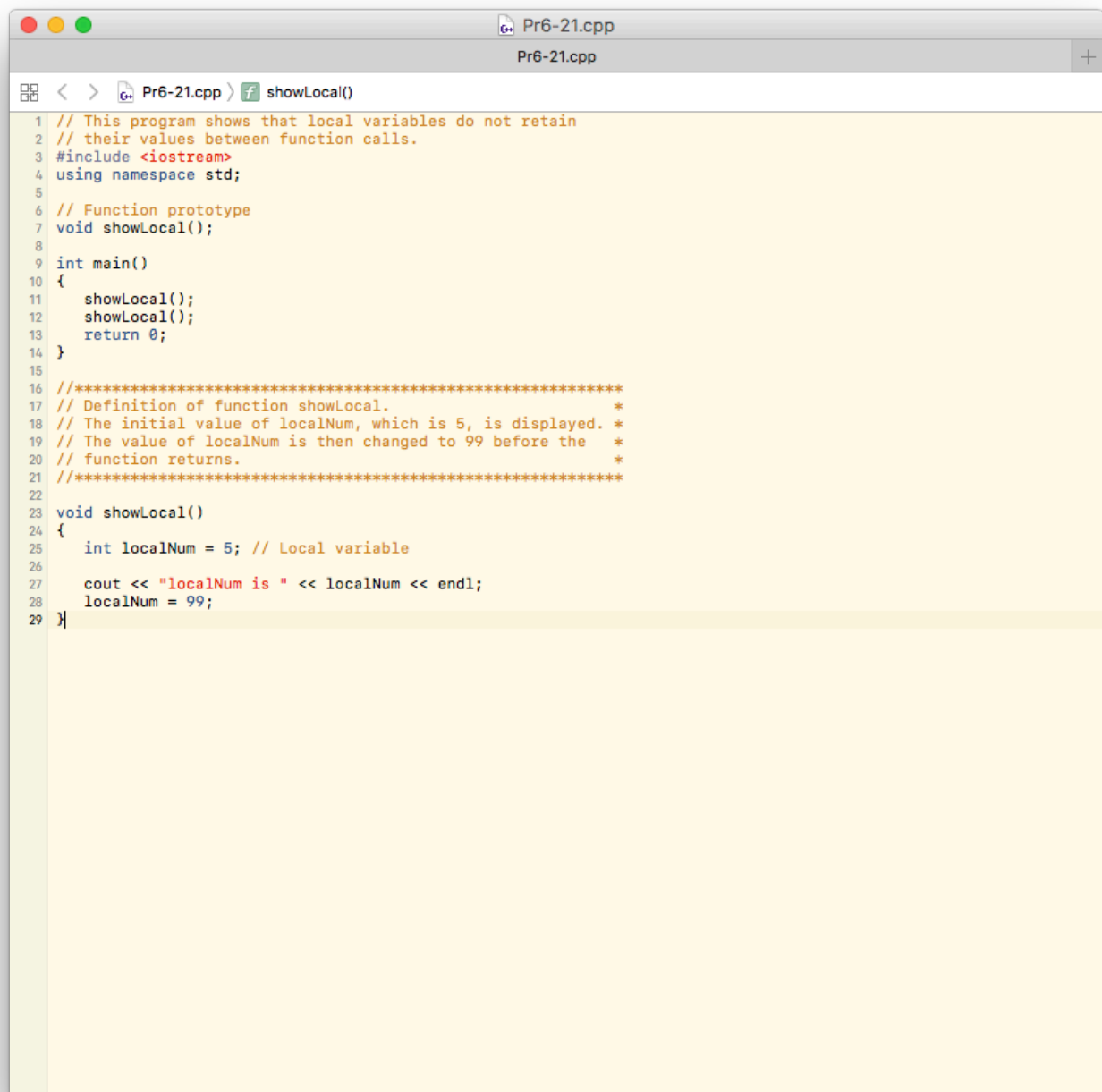
The image shows a screenshot of a C++ IDE window. The window has a title bar with three standard macOS window controls (red, yellow, green buttons) on the left and a title "Pr6-18.cpp" in the center. Below the title bar is a tab bar with a single tab labeled "Pr6-18.cpp" and a "+" button on the right. The main editing area has a light yellow background. At the top of this area is a breadcrumb navigation bar showing a file icon, left and right arrow buttons, the file name "Pr6-18.cpp", and a right-pointing arrow followed by the text "No Selection". The code is written in a monospaced font with syntax highlighting: comments are orange, preprocessor directives and keywords are blue, and string literals are red. The code is as follows:

```
1 // This program has an uninitialized global variable.
2 #include <iostream>
3 using namespace std;
4
5 int globalNum; // Global variable, automatically set to zero
6
7 int main()
8 {
9     cout << "globalNum is " << globalNum << endl;
10    return 0;
11 }
```

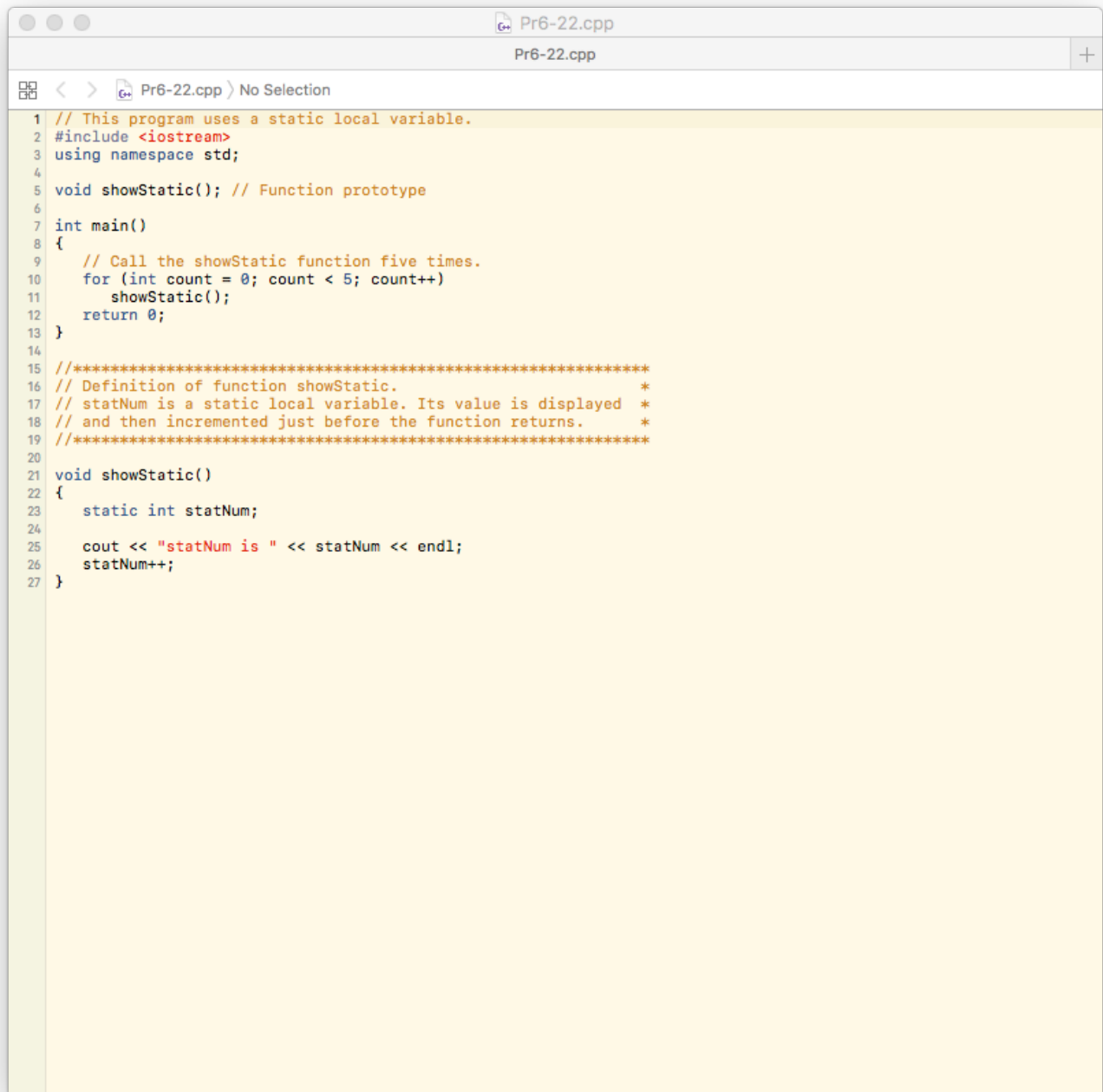
```
Pr6-19.cpp
Pr6-19.cpp
Pr6-19.cpp > No Selection
1 // This program calculates gross pay.
2 #include <iostream>
3 #include <iomanip>
4 using namespace std;
5
6 // Global constants
7 const double PAY_RATE = 22.55; // Hourly pay rate
8 const double BASE_HOURS = 40.0; // Max non-overtime hours
9 const double OT_MULTIPLIER = 1.5; // Overtime multiplier
10
11 // Function prototypes
12 double getBasePay(double);
13 double getOvertimePay(double);
14
15 int main()
16 {
17     double hours, // Hours worked
18           basePay, // Base pay
19           overtime = 0.0, // Overtime pay
20           totalPay; // Total pay
21
22     // Get the number of hours worked.
23     cout << "How many hours did you work? ";
24     cin >> hours;
25
26     // Get the amount of base pay.
27     basePay = getBasePay(hours);
28
29     // Get overtime pay, if any.
30     if (hours > BASE_HOURS)
31         overtime = getOvertimePay(hours);
32
33     // Calculate the total pay.
34     totalPay = basePay + overtime;
35
36     // Set up numeric output formatting.
37     cout << setprecision(2) << fixed << showpoint;
38
39     // Display the pay.
40     cout << "Base pay: $" << basePay << endl
41           << "Overtime pay $" << overtime << endl
42           << "Total pay $" << totalPay << endl;
43     return 0;
44 }
45
46 //*****
47 // The getBasePay function accepts the number of *
48 // hours worked as an argument and returns the *
49 // employee's pay for non-overtime hours. *
50 //*****
51
52 double getBasePay(double hoursWorked)
53 {
54     double basePay; // To hold base pay
55
56     // Determine base pay
```



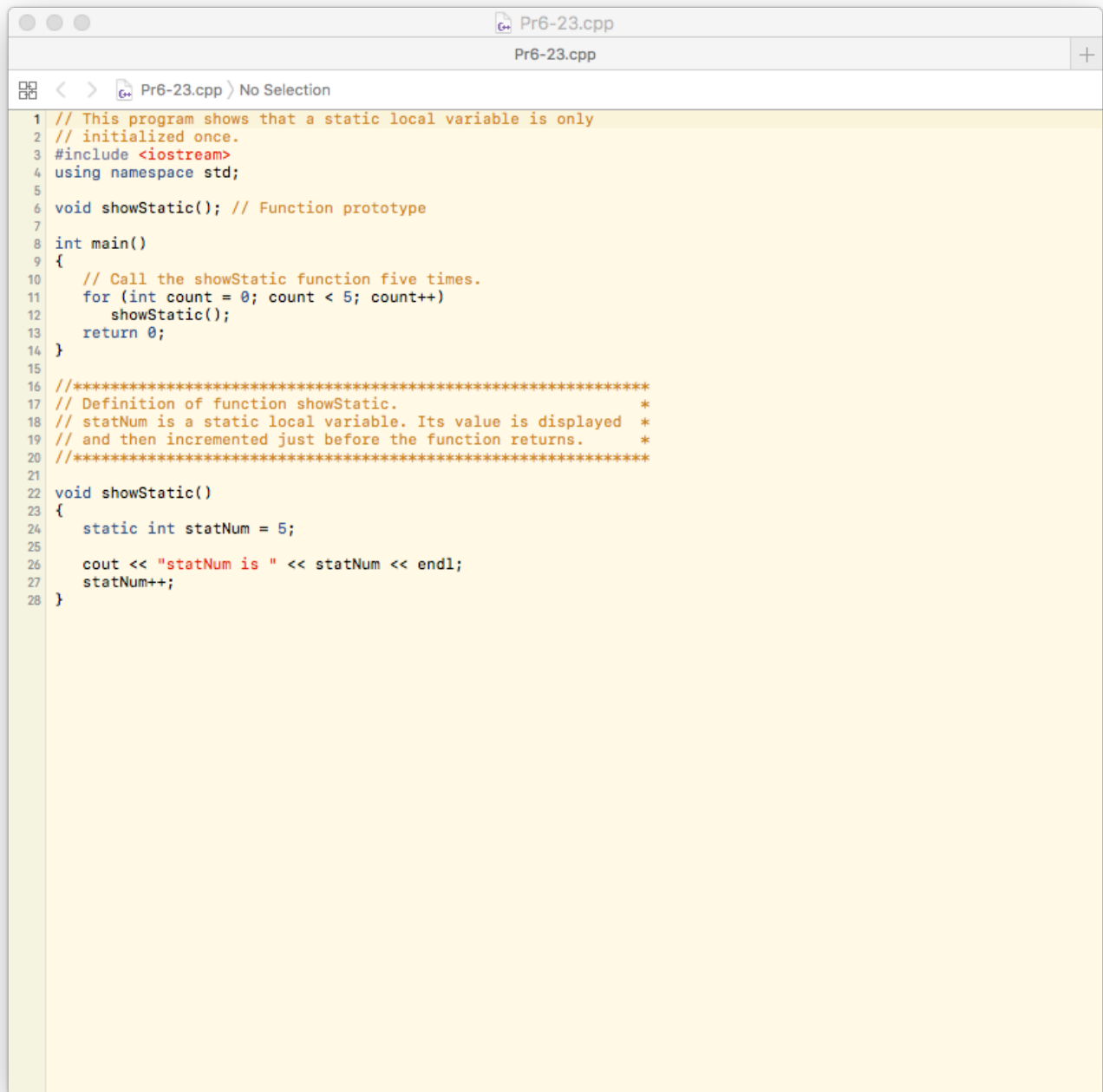
```
1 // This program demonstrates how a local variable
2 // can shadow the name of a global constant.
3 #include <iostream>
4 using namespace std;
5
6 // Global constant.
7 const int BIRDS = 500;
8
9 // Function prototype
10 void california();
11
12 int main()
13 {
14     cout << "In main there are " << BIRDS
15           << " birds.\n";
16     california();
17     return 0;
18 }
19
20 /**
21  * california function
22  */
23
24 void california()
25 {
26     const int BIRDS = 10000;
27     cout << "In california there are " << BIRDS
28           << " birds.\n";
29 }
```



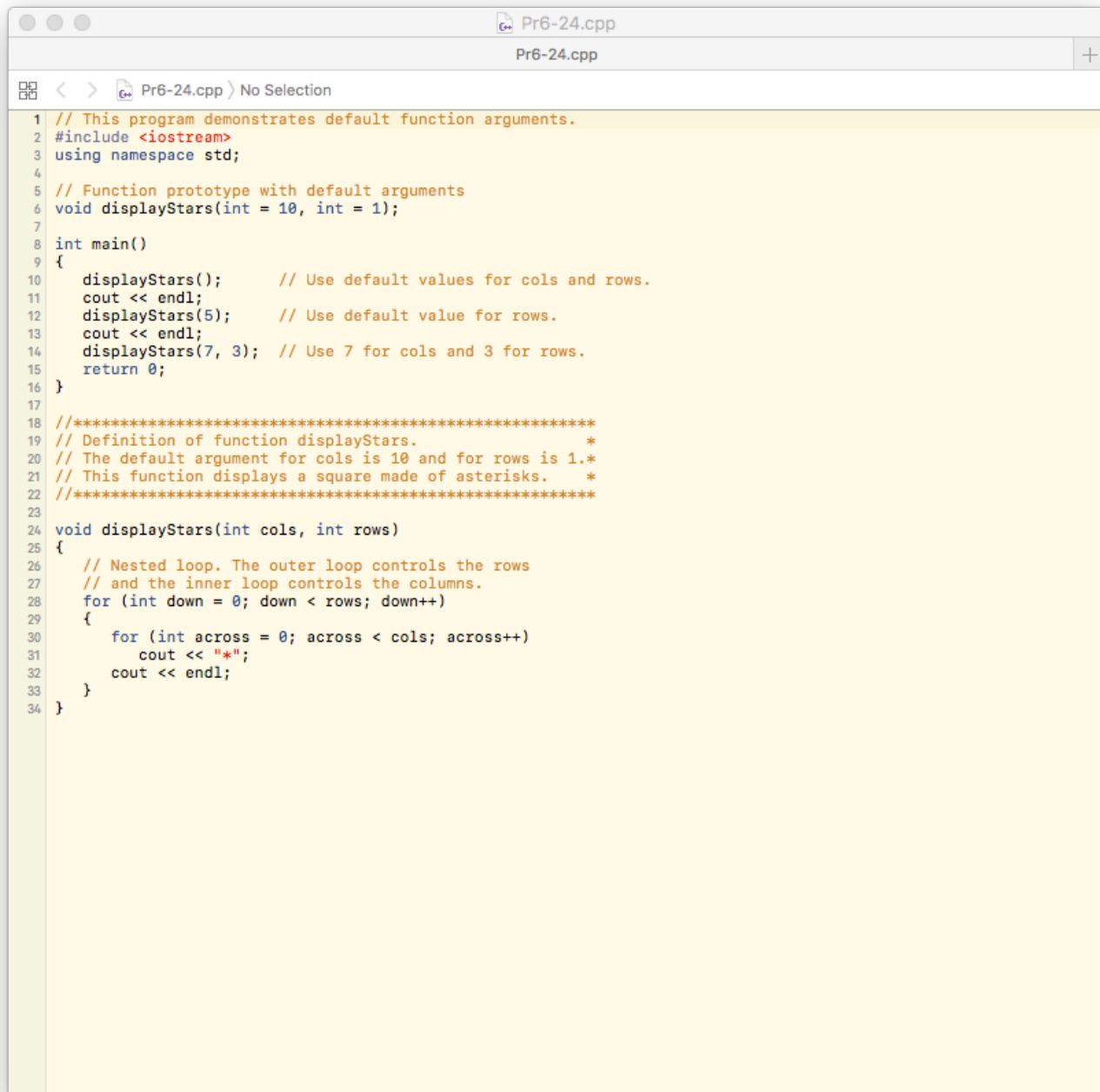
```
1 // This program shows that local variables do not retain
2 // their values between function calls.
3 #include <iostream>
4 using namespace std;
5
6 // Function prototype
7 void showLocal();
8
9 int main()
10 {
11     showLocal();
12     showLocal();
13     return 0;
14 }
15
16 //*****
17 // Definition of function showLocal. *
18 // The initial value of localNum, which is 5, is displayed. *
19 // The value of localNum is then changed to 99 before the *
20 // function returns. *
21 //*****
22
23 void showLocal()
24 {
25     int localNum = 5; // Local variable
26
27     cout << "localNum is " << localNum << endl;
28     localNum = 99;
29 }
```



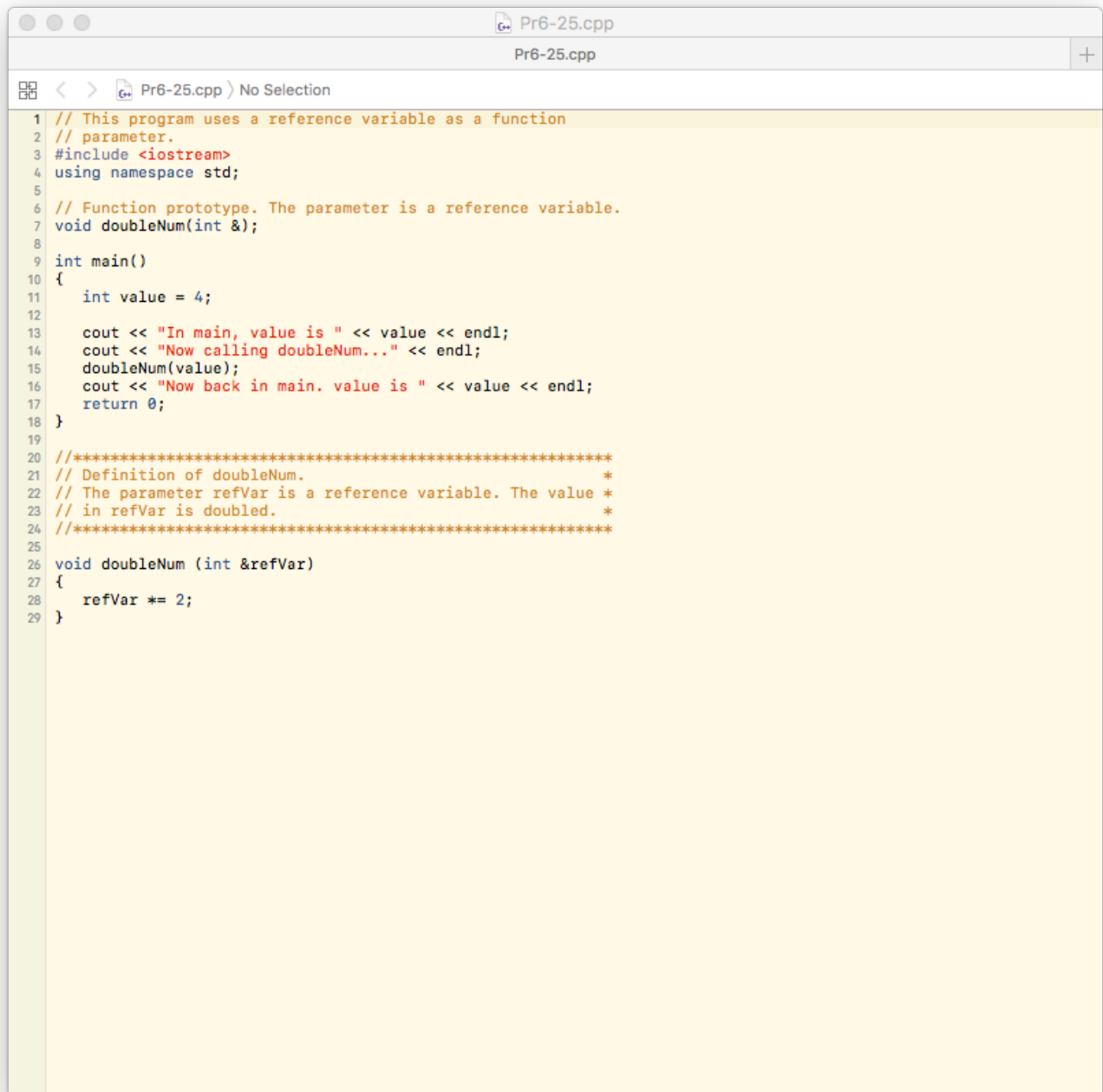
```
1 // This program uses a static local variable.
2 #include <iostream>
3 using namespace std;
4
5 void showStatic(); // Function prototype
6
7 int main()
8 {
9     // Call the showStatic function five times.
10    for (int count = 0; count < 5; count++)
11        showStatic();
12    return 0;
13 }
14
15 //*****
16 // Definition of function showStatic.
17 // statNum is a static local variable. Its value is displayed *
18 // and then incremented just before the function returns.      *
19 //*****
20
21 void showStatic()
22 {
23     static int statNum;
24
25     cout << "statNum is " << statNum << endl;
26     statNum++;
27 }
```



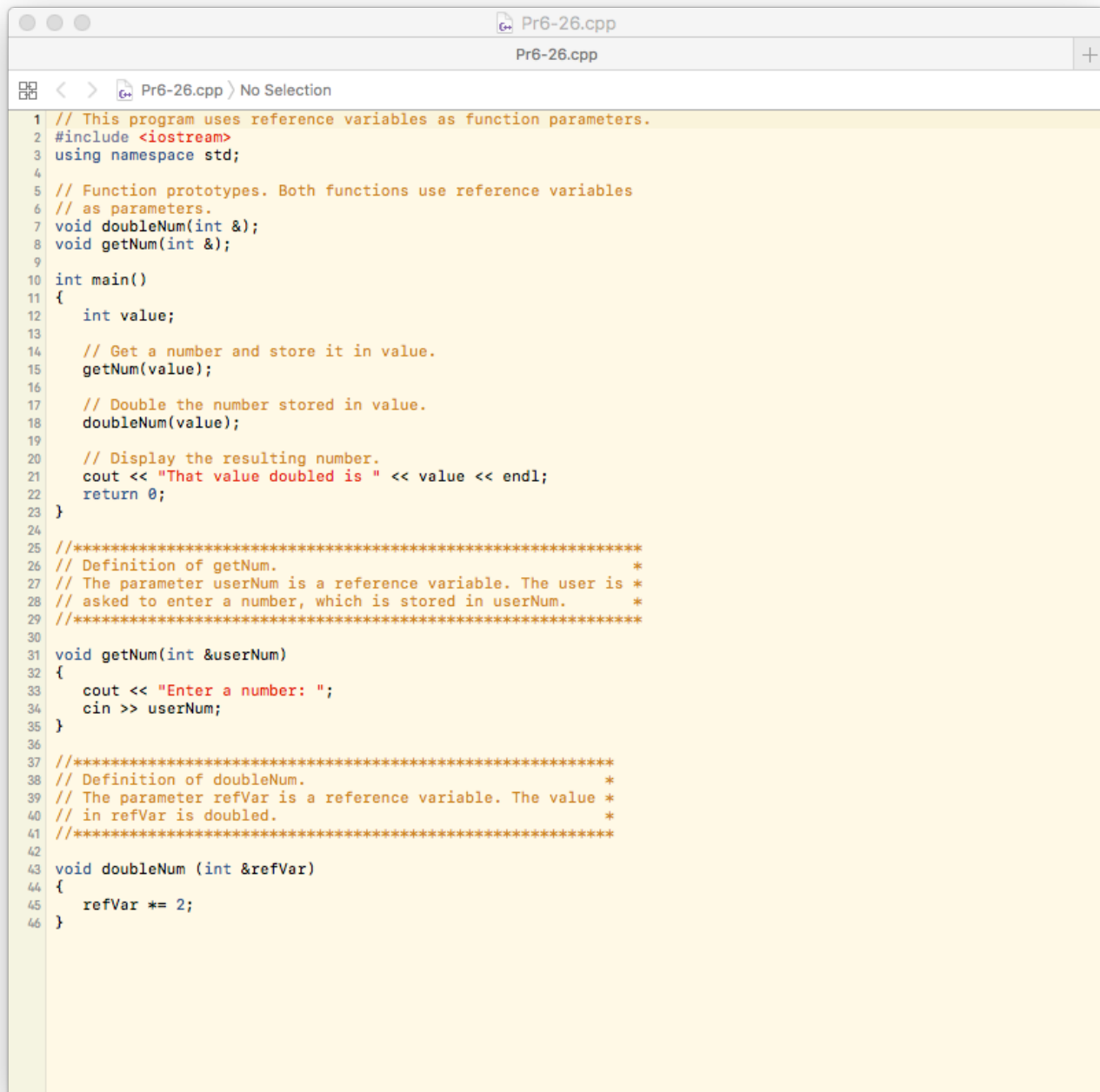
```
1 // This program shows that a static local variable is only
2 // initialized once.
3 #include <iostream>
4 using namespace std;
5
6 void showStatic(); // Function prototype
7
8 int main()
9 {
10     // Call the showStatic function five times.
11     for (int count = 0; count < 5; count++)
12         showStatic();
13     return 0;
14 }
15
16 //*****
17 // Definition of function showStatic.
18 // statNum is a static local variable. Its value is displayed
19 // and then incremented just before the function returns.
20 //*****
21
22 void showStatic()
23 {
24     static int statNum = 5;
25
26     cout << "statNum is " << statNum << endl;
27     statNum++;
28 }
```



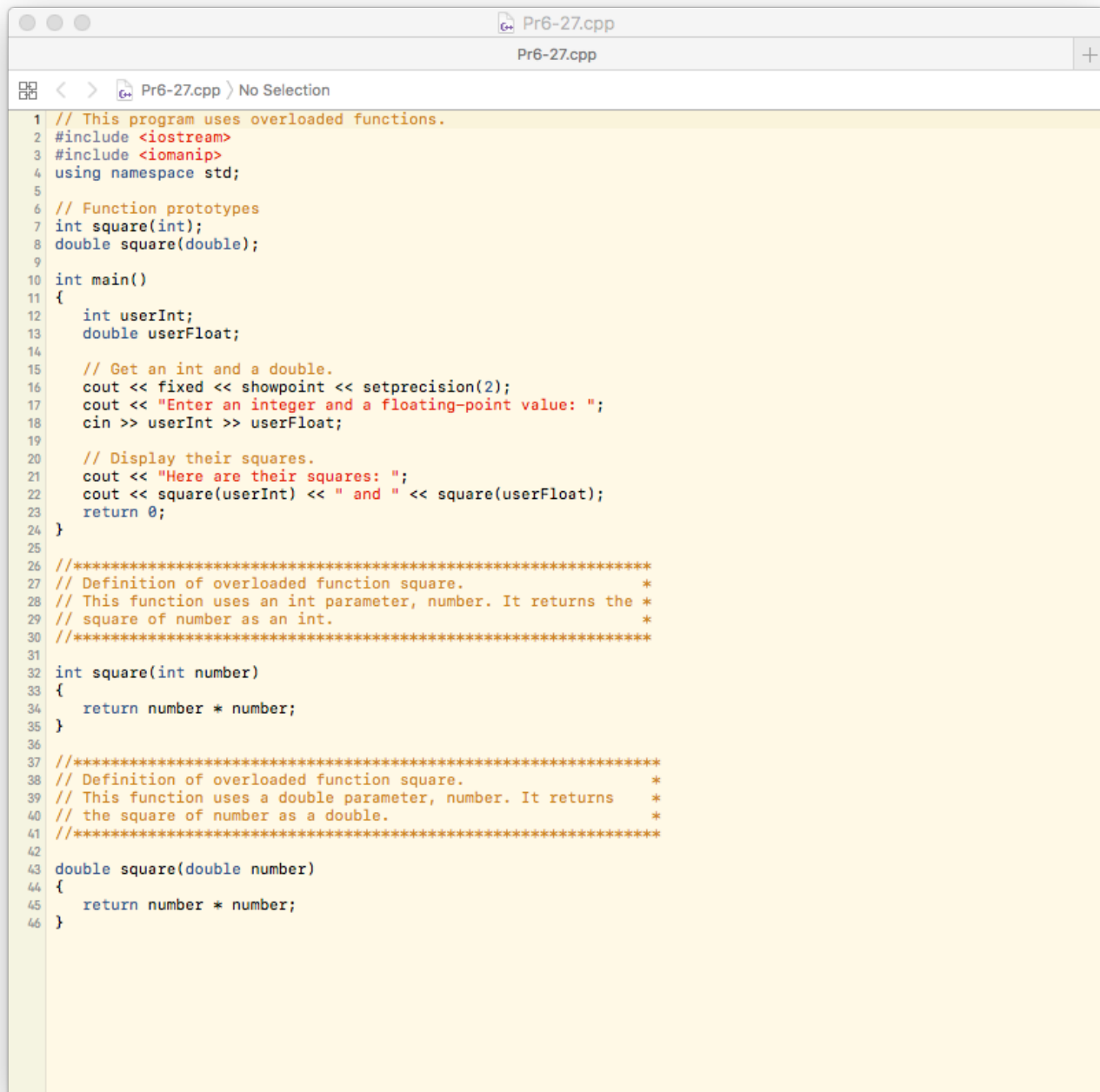
```
1 // This program demonstrates default function arguments.
2 #include <iostream>
3 using namespace std;
4
5 // Function prototype with default arguments
6 void displayStars(int = 10, int = 1);
7
8 int main()
9 {
10     displayStars(); // Use default values for cols and rows.
11     cout << endl;
12     displayStars(5); // Use default value for rows.
13     cout << endl;
14     displayStars(7, 3); // Use 7 for cols and 3 for rows.
15     return 0;
16 }
17
18 //*****
19 // Definition of function displayStars.
20 // The default argument for cols is 10 and for rows is 1.*
21 // This function displays a square made of asterisks. *
22 //*****
23
24 void displayStars(int cols, int rows)
25 {
26     // Nested loop. The outer loop controls the rows
27     // and the inner loop controls the columns.
28     for (int down = 0; down < rows; down++)
29     {
30         for (int across = 0; across < cols; across++)
31             cout << "*";
32         cout << endl;
33     }
34 }
```

```
1 // This program uses a reference variable as a function
2 // parameter.
3 #include <iostream>
4 using namespace std;
5
6 // Function prototype. The parameter is a reference variable.
7 void doubleNum(int &);
8
9 int main()
10 {
11     int value = 4;
12
13     cout << "In main, value is " << value << endl;
14     cout << "Now calling doubleNum..." << endl;
15     doubleNum(value);
16     cout << "Now back in main. value is " << value << endl;
17     return 0;
18 }
19
20 //*****
21 // Definition of doubleNum.
22 // The parameter refVar is a reference variable. The value *
23 // in refVar is doubled.
24 //*****
25
26 void doubleNum (int &refVar)
27 {
28     refVar *= 2;
29 }
```



```
1 // This program uses reference variables as function parameters.
2 #include <iostream>
3 using namespace std;
4
5 // Function prototypes. Both functions use reference variables
6 // as parameters.
7 void doubleNum(int &);
8 void getNum(int &);
9
10 int main()
11 {
12     int value;
13
14     // Get a number and store it in value.
15     getNum(value);
16
17     // Double the number stored in value.
18     doubleNum(value);
19
20     // Display the resulting number.
21     cout << "That value doubled is " << value << endl;
22     return 0;
23 }
24
25 //*****
26 // Definition of getNum.
27 // The parameter userNum is a reference variable. The user is *
28 // asked to enter a number, which is stored in userNum.      *
29 //*****
30
31 void getNum(int &userNum)
32 {
33     cout << "Enter a number: ";
34     cin >> userNum;
35 }
36
37 //*****
38 // Definition of doubleNum.
39 // The parameter refVar is a reference variable. The value *
40 // in refVar is doubled.
41 //*****
42
43 void doubleNum (int &refVar)
44 {
45     refVar *= 2;
46 }
```



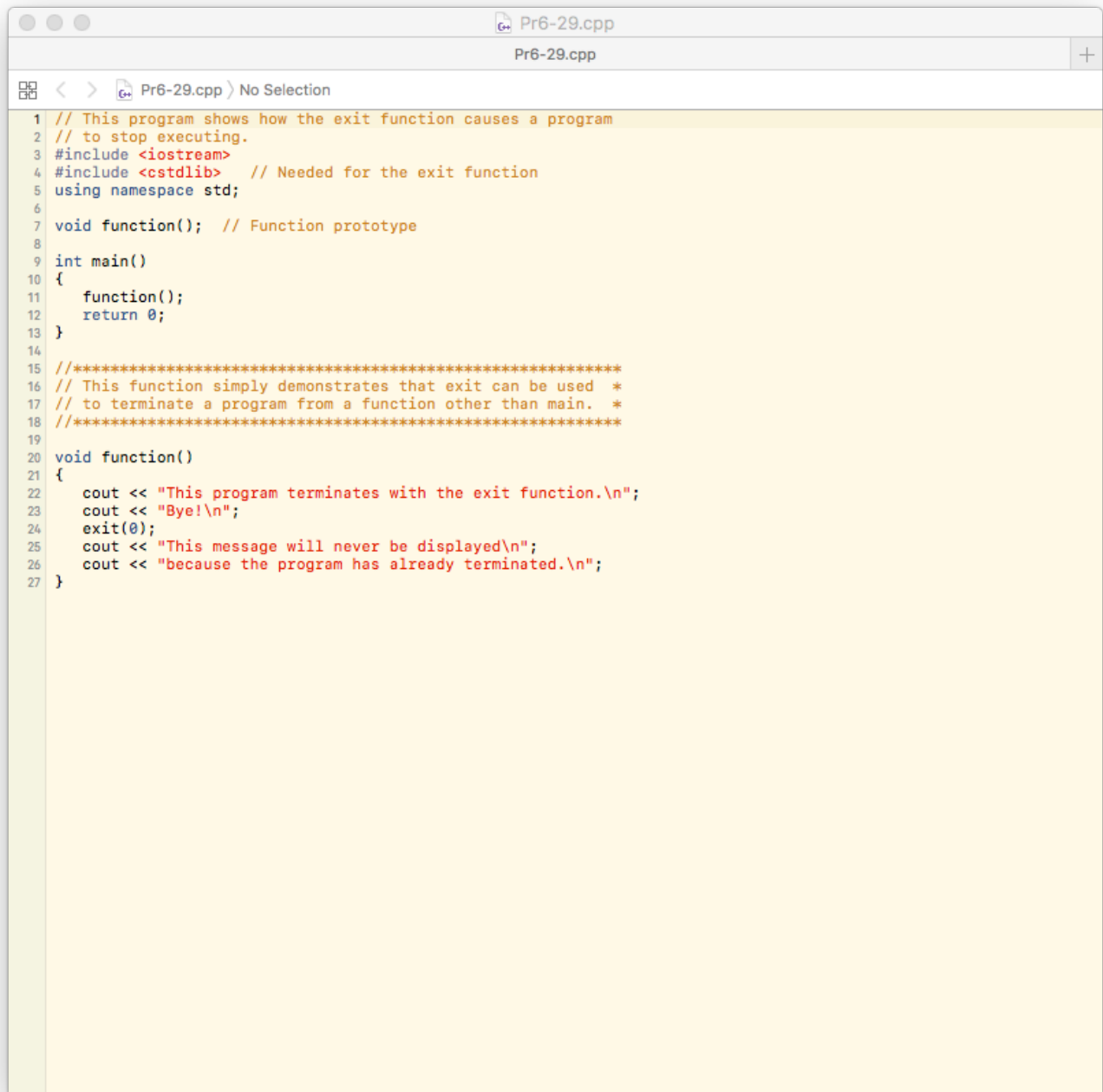
The image shows a screenshot of a C++ IDE window titled "Pr6-27.cpp". The window has a standard macOS-style title bar with three colored buttons (red, yellow, green) on the left. Below the title bar is a toolbar with icons for file operations and a search icon. The main area of the window displays the source code for a C++ program. The code is as follows:

```
1 // This program uses overloaded functions.
2 #include <iostream>
3 #include <iomanip>
4 using namespace std;
5
6 // Function prototypes
7 int square(int);
8 double square(double);
9
10 int main()
11 {
12     int userInt;
13     double userFloat;
14
15     // Get an int and a double.
16     cout << fixed << showpoint << setprecision(2);
17     cout << "Enter an integer and a floating-point value: ";
18     cin >> userInt >> userFloat;
19
20     // Display their squares.
21     cout << "Here are their squares: ";
22     cout << square(userInt) << " and " << square(userFloat);
23     return 0;
24 }
25
26 //*****
27 // Definition of overloaded function square.
28 // This function uses an int parameter, number. It returns the
29 // square of number as an int.
30 //*****
31
32 int square(int number)
33 {
34     return number * number;
35 }
36
37 //*****
38 // Definition of overloaded function square.
39 // This function uses a double parameter, number. It returns
40 // the square of number as a double.
41 //*****
42
43 double square(double number)
44 {
45     return number * number;
46 }
```

```
Pr6-28.cpp
Pr6-28.cpp

Pr6-28.cpp No Selection

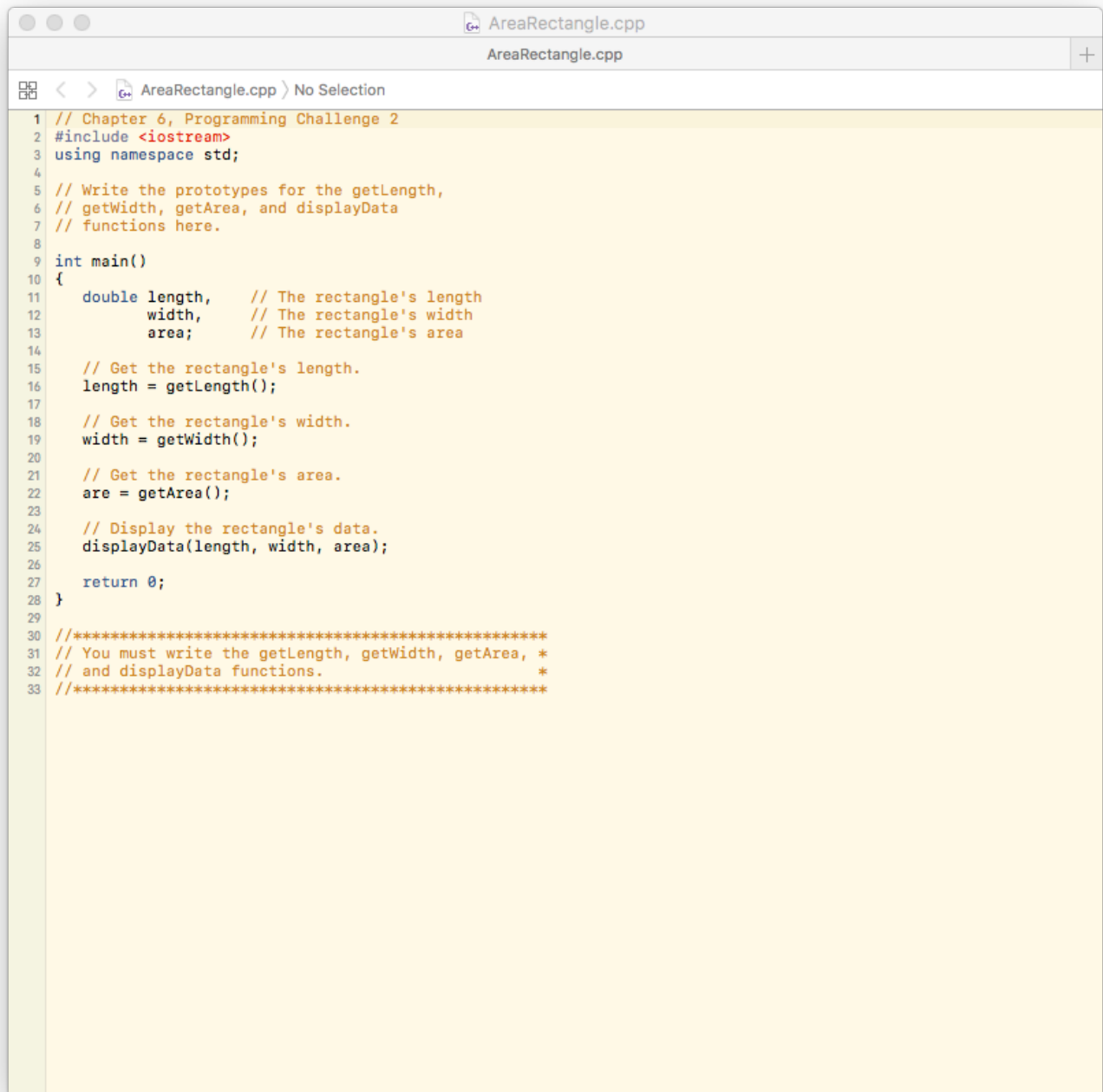
1 // This program demonstrates overloaded functions to calculate
2 // the gross weekly pay of hourly-paid or salaried employees.
3 #include <iostream>
4 #include <iomanip>
5 using namespace std;
6
7 // Function prototypes
8 void getChoice(char &);
9 double calcWeeklyPay(int, double);
10 double calcWeeklyPay(double);
11
12 int main()
13 {
14     char selection; // Menu selection
15     int worked; // Hours worked
16     double rate; // Hourly pay rate
17     double yearly; // Yearly salary
18
19     // Set numeric output formatting.
20     cout << fixed << showpoint << setprecision(2);
21
22     // Display the menu and get a selection.
23     cout << "Do you want to calculate the weekly pay of\n";
24     cout << "(H) an hourly-paid employee, or \n";
25     cout << "(S) a salaried employee?\n";
26     getChoice(selection);
27
28     // Process the menu selection.
29     switch (selection)
30     {
31         // Hourly-paid employee
32         case 'H' :
33         case 'h' : cout << "How many hours were worked? ";
34                     cin >> worked;
35                     cout << "What is the hour pay rate? ";
36                     cin >> rate;
37                     cout << "The gross weekly pay is $";
38                     cout << calcWeeklyPay(worked, rate) << endl;
39                     break;
40
41         // Salaried employee
42         case 'S' :
43         case 's' : cout << "What is the annual salary? ";
44                     cin >> yearly;
45                     cout << "The gross weekly pay is $";
46                     cout << calcWeeklyPay(yearly) << endl;
47                     break;
48     }
49     return 0;
50 }
51
52 //*****
53 // Definition of function getChoice. *
54 // The parameter letter is a reference to a char. *
55 // This function asks the user for an H or an S and returns *
56 // the validated input. *
57 //*****
58
59 void getChoice(char &letter)
60 {
61     // Get the user's selection.
62     cout << "Enter your choice (H or S): ";
63     cin >> letter;
64
65     // Validate the selection.
66     while (letter != 'H' && letter != 'h' &&
67            letter != 'S' && letter != 's')
68     {
69         cout << "Please enter H or S: ";
70         cin >> letter;
71     }
72 }
73
74 //*****
75 // Definition of overloaded function calcWeeklyPay. *
76 // This function calculates the gross weekly pay of *
77 // an hourly-paid employee. The parameter hours holds the *
78 // number of hours worked. The parameter payRate holds the *
79 // hourly pay rate. The function returns the weekly salary. *
80 //*****
81
82 double calcWeeklyPay(int hours, double payRate)
83 {
84     return hours * payRate;
85 }
86
87 //*****
88 // Definition of overloaded function calcWeeklyPay. *
89 // This function calculates the gross weekly pay of *
90 // a salaried employee. The parameter holds the employee's *
91 // annual salary. The function returns the weekly salary. *
92 //*****
93
94 double calcWeeklyPay(double annSalary)
95 {
96     return annSalary / 52;
97 }
```



```
1 // This program shows how the exit function causes a program
2 // to stop executing.
3 #include <iostream>
4 #include <cstdlib> // Needed for the exit function
5 using namespace std;
6
7 void function(); // Function prototype
8
9 int main()
10 {
11     function();
12     return 0;
13 }
14
15 //*****
16 // This function simply demonstrates that exit can be used *
17 // to terminate a program from a function other than main. *
18 //*****
19
20 void function()
21 {
22     cout << "This program terminates with the exit function.\n";
23     cout << "Bye!\n";
24     exit(0);
25     cout << "This message will never be displayed\n";
26     cout << "because the program has already terminated.\n";
27 }
```

```
Pr6-30.cpp
Pr6-30.cpp
Pr6-30.cpp No Selection

1 // This program is a driver for testing the showFees function.
2 #include <iostream>
3 using namespace std;
4
5 // Prototype
6 void showFees(double, int);
7
8 int main()
9 {
10     // Constants for membership rates
11     const double ADULT = 40.0;
12     const double SENIOR = 30.0;
13     const double CHILD = 20.0;
14
15     // Perform a test for adult membership.
16     cout << "Testing an adult membership...\n"
17          << "Calling the showFees function with arguments "
18          << ADULT << " and 10.\n";
19     showFees(ADULT, 10);
20
21     // Perform a test for senior citizen membership.
22     cout << "\nTesting a senior citizen membership...\n"
23          << "Calling the showFees function with arguments "
24          << SENIOR << " and 10.\n";
25     showFees(SENIOR, 10);
26
27     // Perform a test for child membership.
28     cout << "\nTesting a child membership...\n"
29          << "Calling the showFees function with arguments "
30          << CHILD << " and 10.\n";
31     showFees(CHILD, 10);
32     return 0;
33 }
34
35 //*****
36 // Definition of function showFees. The memberRate parameter    *
37 // the monthly membership rate and the months parameter holds the *
38 // number of months. The function displays the total charges.    *
39 //*****
40
41 void showFees(double memberRate, int months)
42 {
43     cout << "The total charges are $"
44          << (memberRate * months) << endl;
45 }
```



```
1 // Chapter 6, Programming Challenge 2
2 #include <iostream>
3 using namespace std;
4
5 // Write the prototypes for the getLength,
6 // getWidth, getArea, and displayData
7 // functions here.
8
9 int main()
10 {
11     double length,    // The rectangle's length
12           width,      // The rectangle's width
13           area;       // The rectangle's area
14
15     // Get the rectangle's length.
16     length = getLength();
17
18     // Get the rectangle's width.
19     width = getWidth();
20
21     // Get the rectangle's area.
22     are = getArea();
23
24     // Display the rectangle's data.
25     displayData(length, width, area);
26
27     return 0;
28 }
29
30 //*****
31 // You must write the getLength, getWidth, getArea, *
32 // and displayData functions. *
33 //*****
```

HealthClubWithStub.cpp > No Selection

```

1 // This is a menu-driven program that makes a function call
2 // for each selection the user makes.
3 #include <iostream>
4 #include <iomanip>
5 using namespace std;
6
7 // Function prototypes
8 void showMenu();
9 void showFees(double, int);
10
11 int main()
12 {
13     int choice;          // To hold a menu choice
14     int months;          // To hold a number of months
15
16     // Constants for membership rates
17     const double ADULT = 40.0;
18     const double SENIOR = 30.0;
19     const double CHILD = 20.0;
20
21     // Set up numeric output formatting.
22     cout << fixed << showpoint << setprecision(2);
23
24     do
25     {
26         // Display the menu and get the user's choice.
27         showMenu();
28         cin >> choice;
29
30         // Validate the menu selection.
31         while (choice < 1 || choice > 4)
32         {
33             cout << "Please enter 1, 2, 3, or 4: ";
34             cin >> choice;
35         }
36
37         if (choice != 4)
38         {
39             // Get the number of months.
40             cout << "For how many months? ";
41             cin >> months;
42
43             // Display the membership fees.
44             switch (choice)
45             {
46                 case 1: showFees(ADULT, months);
47                         break;
48                 case 2: showFees(CHILD, months);
49                         break;
50                 case 3: showFees(SENIOR, months);
51
52             }
53         } while (choice != 4);
54         return 0;
55     }
56
57     //*****
58     // Definition of function showMenu which displays the menu.      *
59     //*****
60
61     void showMenu()
62     {
63         cout << "\n\t\tHealth Club Membership Menu\n\n";
64         cout << "1. Standard Adult Membership\n";
65         cout << "2. Child Membership\n";
66         cout << "3. Senior Citizen Membership\n";
67         cout << "4. Quit the Program\n\n";
68         cout << "Enter your choice: ";
69     }
70
71     //*****
72     // showFee stub      *
73     //*****
74
75     void showFees(double memberRate, int months)
76     {
77         cout << "The showFees function was called with "
78              << "the following arguments:\n"
79              << "memberRate: " << memberRate << endl
80              << "months: " << months << endl;
81     }
82
83

```