Pr6-1.cpp

Pr6-1.cpp

Pr6-1.cpp 〉 No Selection

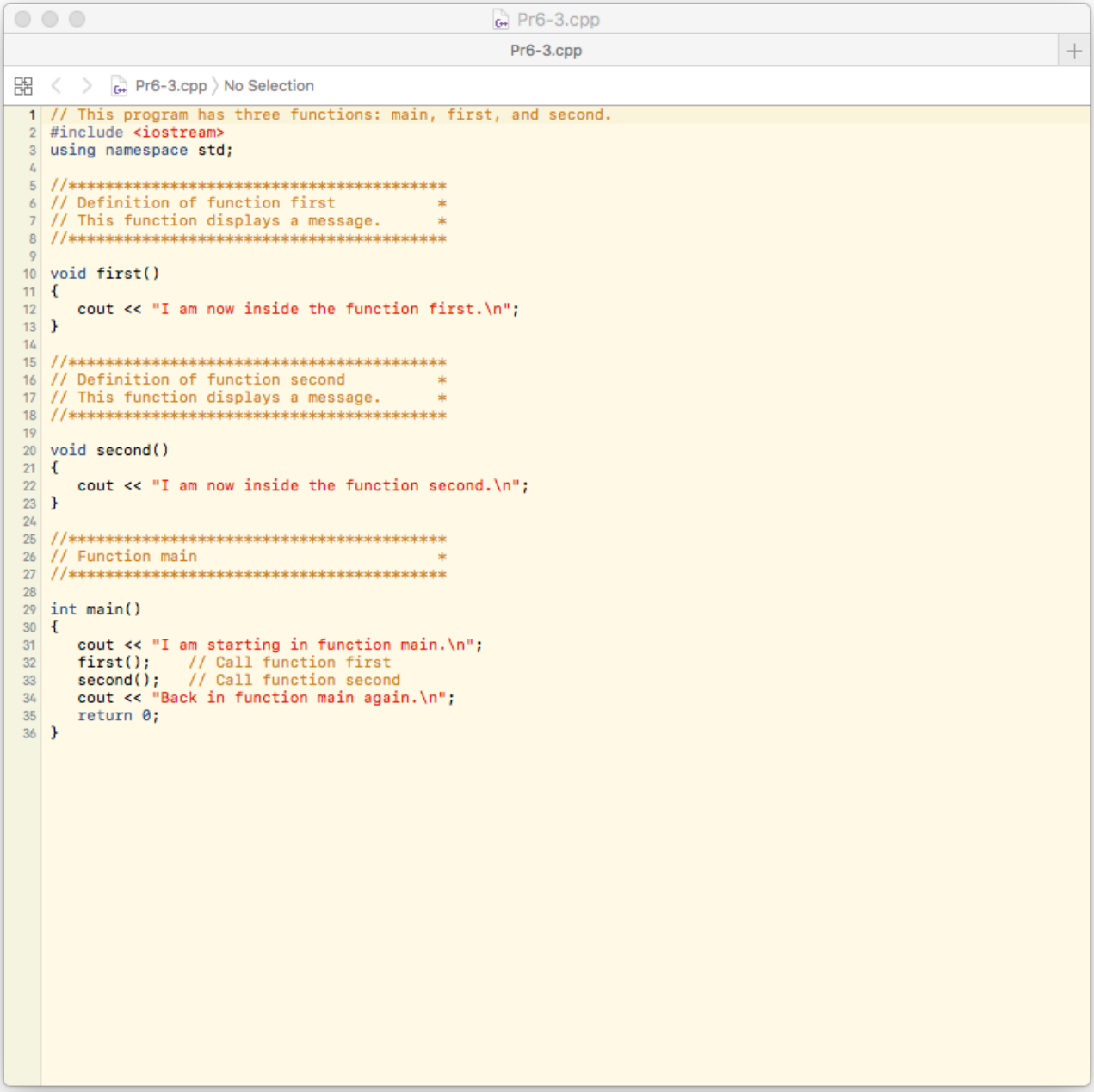
```cpp
// This program has two functions: main and displayMessage
#include <iostream>
using namespace std;

//*****************************************
// Definition of function displayMessage  *
// This function displays a greeting.     *
//*****************************************

void displayMessage()
{
   cout << "Hello from the function displayMessage.\n";
}

//*****************************************
// Function main                          *
//*****************************************

int main()
{
   cout << "Hello from main.\n";
   displayMessage();
   cout << "Back in function main again.\n";
   return 0;
}
```

Pr6-2.cpp

Pr6-2.cpp

Pr6-2.cpp 〉 No Selection

```cpp
// The function displayMessage is repeatedly called from a loop.
#include <iostream>
using namespace std;

//****************************************
// Definition of function displayMessage  *
// This function displays a greeting.     *
//****************************************

void displayMessage()
{
   cout << "Hello from the function displayMessage.\n";
}

//****************************************
// Function main                          *
//****************************************

int main()
{
   cout << "Hello from main.\n";
   for (int count = 0; count < 5; count++)
      displayMessage(); // Call displayMessage
   cout << "Back in function main again.\n";
   return 0;
}
```


Pr6-3.cpp
Pr6-3.cpp
Pr6-3.cpp 〉 No Selection
// This program has three functions: main, first, and second.
#include <iostream>
using namespace std;
//***
// Definition of function first   *
// This function displays a message.   *
//***
void first()
{
cout << "I am now inside the function first.\n";
}
//***
// Definition of function second   *
// This function displays a message.   *
//***
void second()
{
cout << "I am now inside the function second.\n";
}
//***
// Function main   *
//***
int main()
{
cout << "I am starting in function main.\n";
first();   // Call function first
second();   // Call function second
cout << "Back in function main again.\n";
return 0;
}

Pr6-4.cpp

Pr6-4.cpp

Pr6-4.cpp 〉 No Selection

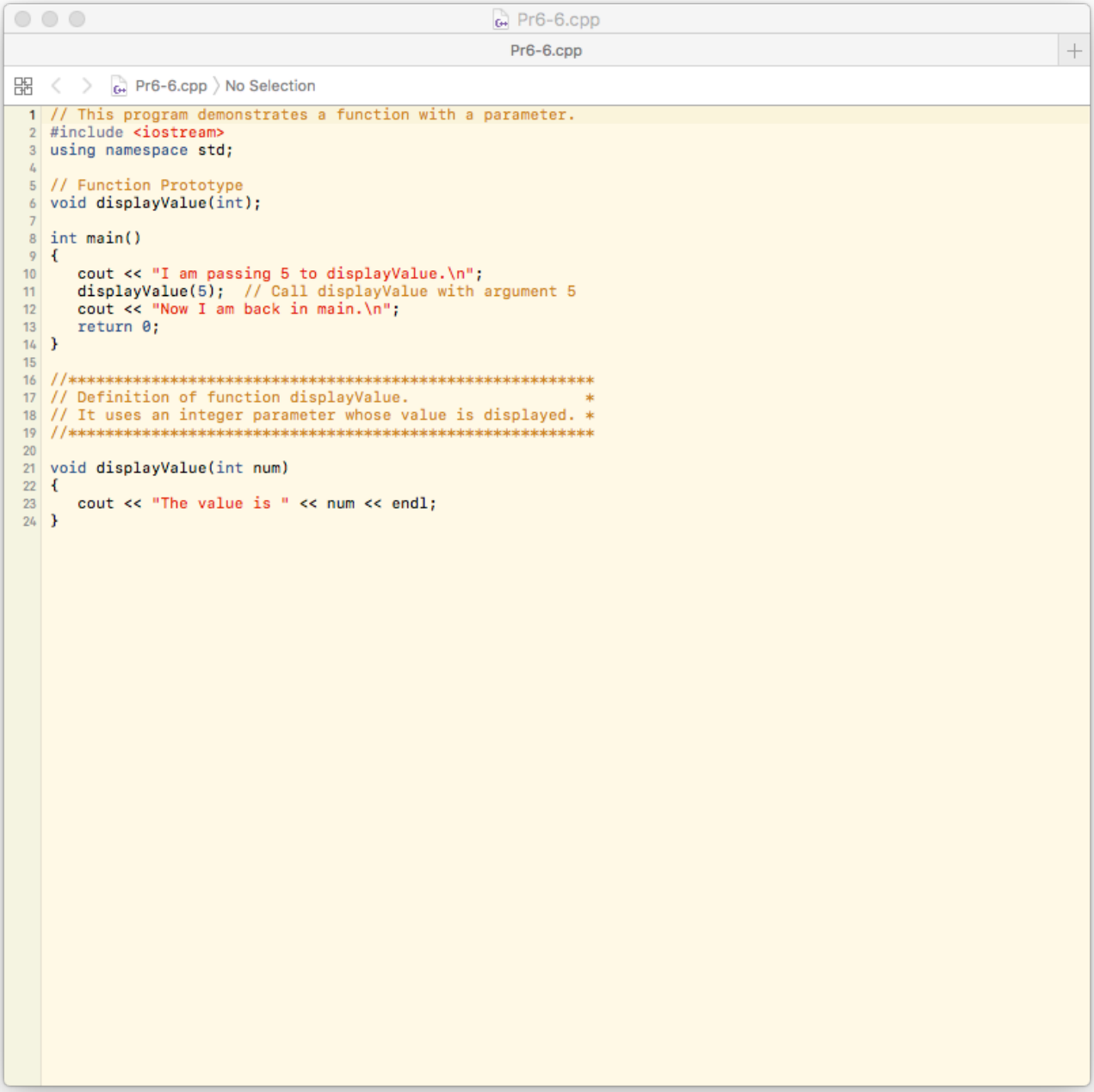
```cpp
// This program has three functions: main, deep, and deeper
#include <iostream>
using namespace std;

//*****************************************
// Definition of function deeper          *
// This function displays a message.      *
//*****************************************

void deeper()
{
   cout << "I am now inside the function deeper.\n";
}

//*****************************************
// Definition of function deep            *
// This function displays a message.      *
//*****************************************

void deep()
{
   cout << "I am now inside the function deep.\n";
   deeper();   // Call function deeper
   cout << "Now I am back in deep.\n";
}

//*****************************************
// Function main                          *
//*****************************************

int main()
{
   cout << "I am starting in function main.\n";
   deep();     // Call function deep
   cout << "Back in function main again.\n";
   return 0;
}
```

Pr6-5.cpp

Pr6-5.cpp

Pr6-5.cpp 〉 No Selection

```cpp
// This program has three functions: main, first, and second.
#include <iostream>
using namespace std;

// Function Prototypes
void first();
void second();

int main()
{
   cout << "I am starting in function main.\n";
   first();    // Call function first
   second();   // Call function second
   cout << "Back in function main again.\n";
   return 0;
}

//*************************************
// Definition of function first.      *
// This function displays a message.  *
//*************************************

void first()
{
   cout << "I am now inside the function first.\n";
}

//*************************************
// Definition of function second.     *
// This function displays a message.  *
//*************************************

void second()
{
   cout << "I am now inside the function second.\n";
}
```


Pr6-6.cpp
Pr6-6.cpp
Pr6-6.cpp 〉 No Selection
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 }

Pr6-7.cpp

Pr6-7.cpp

Pr6-7.cpp 〉 No Selection

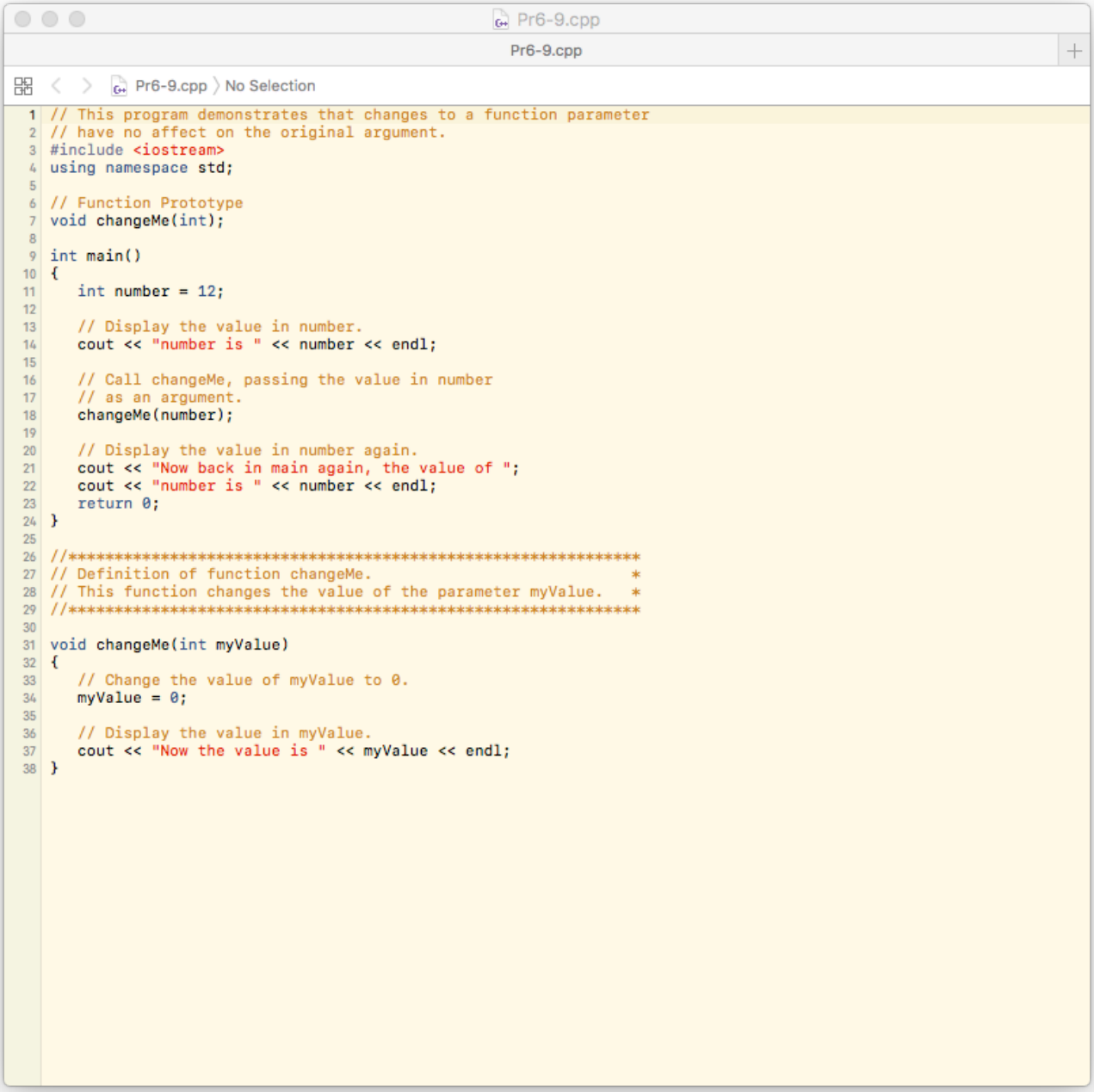
```cpp
// This program demonstrates a function with a parameter.
#include <iostream>
using namespace std;

// Function Prototype
void displayValue(int);

int main()
{
   cout << "I am passing several values to displayValue.\n";
   displayValue(5);  // Call displayValue with argument 5
   displayValue(10); // Call displayValue with argument 10
   displayValue(2);  // Call displayValue with argument 2
   displayValue(16); // Call displayValue with argument 16
   cout << "Now I am back in main.\n";
   return 0;
}

//*********************************************************
// Definition of function displayValue.                   *
// It uses an integer parameter whose value is displayed. *
//*********************************************************

void displayValue(int num)
{
   cout << "The value is " << num << endl;
}
```

Pr6-8.cpp

Pr6-8.cpp

Pr6-8.cpp 〉 No Selection

```cpp
// This program demonstrates a function with three parameters.
#include <iostream>
using namespace std;

// Function Prototype
void showSum(int, int, int);

int main()
{
   int value1, value2, value3;

   // Get three integers.
   cout << "Enter three integers and I will display ";
   cout << "their sum: ";
   cin >> value1 >> value2 >> value3;

   // Call showSum passing three arguments.
   showSum(value1, value2, value3);
   return 0;
}

//*************************************************************
// Definition of function showSum.                            *
// It uses three integer parameters. Their sum is displayed. *
//*************************************************************

void showSum(int num1, int num2, int num3)
{
   cout << (num1 + num2 + num3) << endl;
}
```


Pr6-9.cpp
Pr6-9.cpp
Pr6-9.cpp 〉 No Selection
// This program demonstrates that changes to a function parameter
// have no affect on the original argument.
#include <iostream>
using namespace std;
// Function Prototype
void changeMe(int);
int main()
{
int number = 12;
// Display the value in number.
cout << "number is " << number << endl;
// Call changeMe, passing the value in number
// as an argument.
changeMe(number);
// Display the value in number again.
cout << "Now back in main again, the value of ";
cout << "number is " << number << endl;
return 0;
}
//***
// Definition of function changeMe. *
// This function changes the value of the parameter myValue. *
//***
void changeMe(int myValue)
{
// Change the value of myValue to 0.
myValue = 0;
// Display the value in myValue.
cout << "Now the value is " << myValue << endl;
}

Pr6-10.cpp

Pr6-10.cpp

Pr6-10.cpp 〉 No Selection

```cpp
// This is a menu-driven program that makes a function call
// for each selection the user makes.
#include <iostream>
#include <iomanip>
using namespace std;

// Function prototypes
void showMenu();
void showFees(double, int);

int main()
{
   int choice;       // To hold a menu choice
   int months;       // To hold a number of months

   // Constants for the menu choices
   const int ADULT_CHOICE = 1,
             CHILD_CHOICE = 2,
             SENIOR_CHOICE = 3,
             QUIT_CHOICE = 4;

   // Constants for membership rates
   const double ADULT = 40.0,
                SENIOR = 30.0,
                CHILD = 20.0;

   // Set up numeric output formatting.
   cout << fixed << showpoint << setprecision(2);

   do
   {
      // Display the menu and get the user's choice.
      showMenu();
      cin >> choice;

      // Validate the menu selection.
      while (choice < ADULT_CHOICE || choice > QUIT_CHOICE)
      {
         cout << "Please enter a valid menu choice: ";
         cin >> choice;
      }

      // If the user does not want to quit, proceed.
      if (choice != QUIT_CHOICE)
      {
          // Get the number of months.
          cout << "For how many months? ";
          cin >> months;

          // Display the membership fees.
          switch (choice)
          {
             case ADULT_CHOICE:
                  showFees(ADULT, months);
                  break;
             case CHILD_CHOICE:
```

Pr6-11.cpp

Pr6-11.cpp

Pr6-11.cpp 〉 No Selection

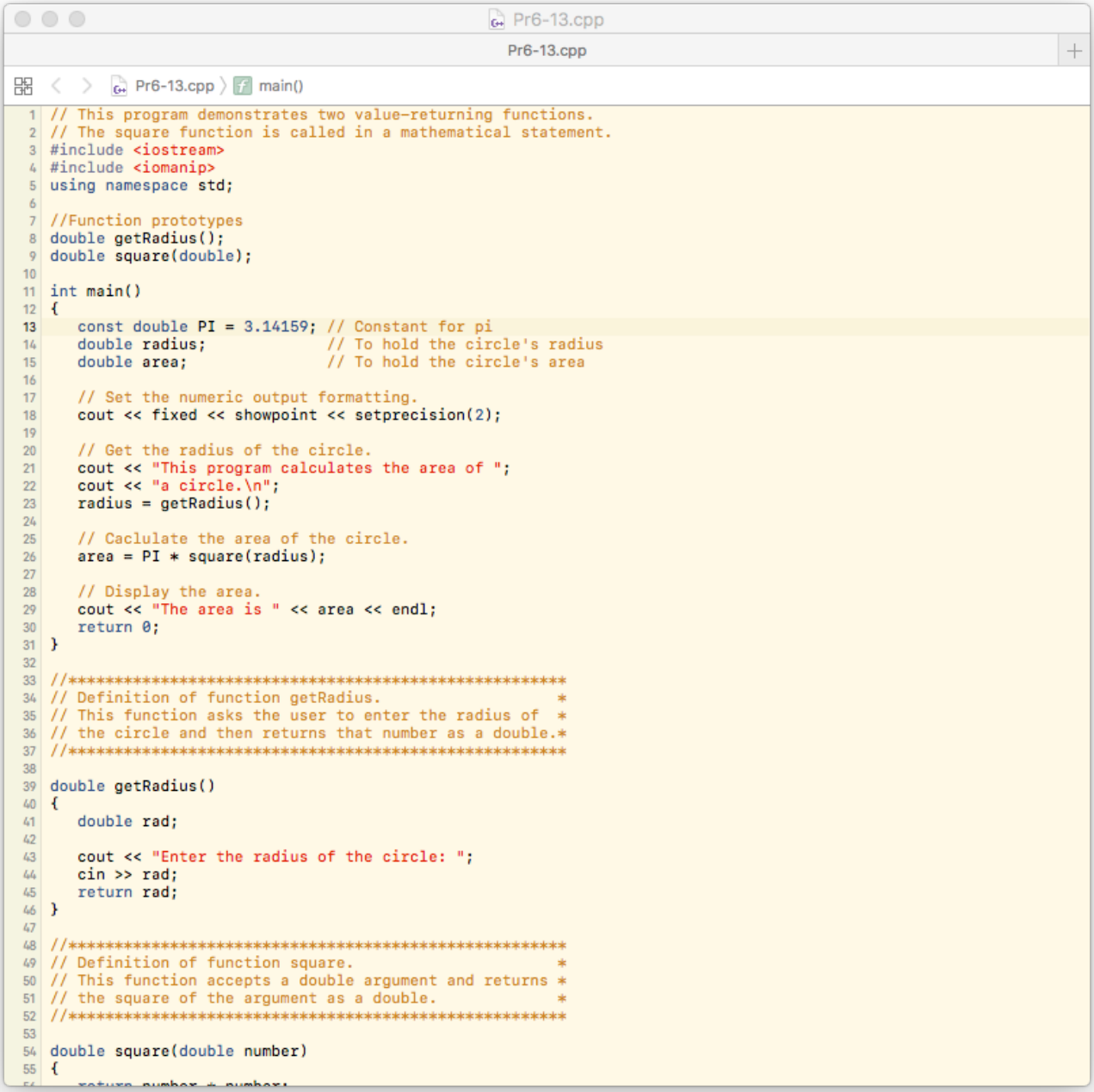
```cpp
// This program uses a function to perform division. If division
// by zero is detected, the function returns.
#include <iostream>
using namespace std;

// Function prototype.
void divide(double, double);

int main()
{
   double num1, num2;

   cout << "Enter two numbers and I will divide the first\n";
   cout << "number by the second number: ";
   cin >> num1 >> num2;
   divide(num1, num2);
   return 0;
}

//*************************************************************
// Definition of function divide.                             *
// Uses two parameters: arg1 and arg2. The function divides arg1*
// by arg2 and shows the result. If arg2 is zero, however, the  *
// function returns.                                          *
//*************************************************************

void divide(double arg1, double arg2)
{
   if (arg2 == 0.0)
   {
      cout << "Sorry, I cannot divide by zero.\n";
      return;
   }
   cout << "The quotient is " << (arg1 / arg2) << endl;
}
```

Pr6-12.cpp

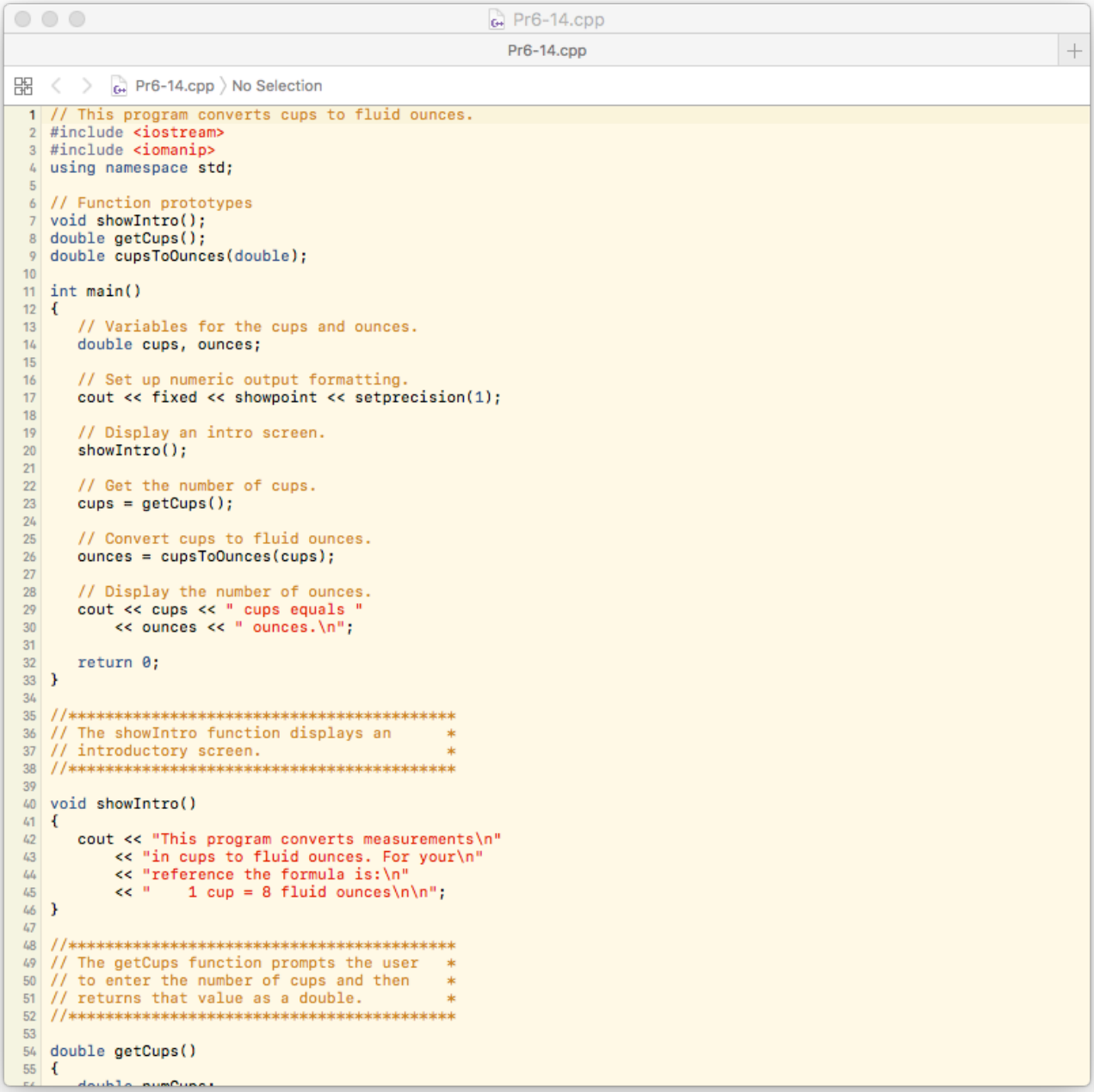
Pr6-12.cpp

Pr6-12.cpp 〉 No Selection

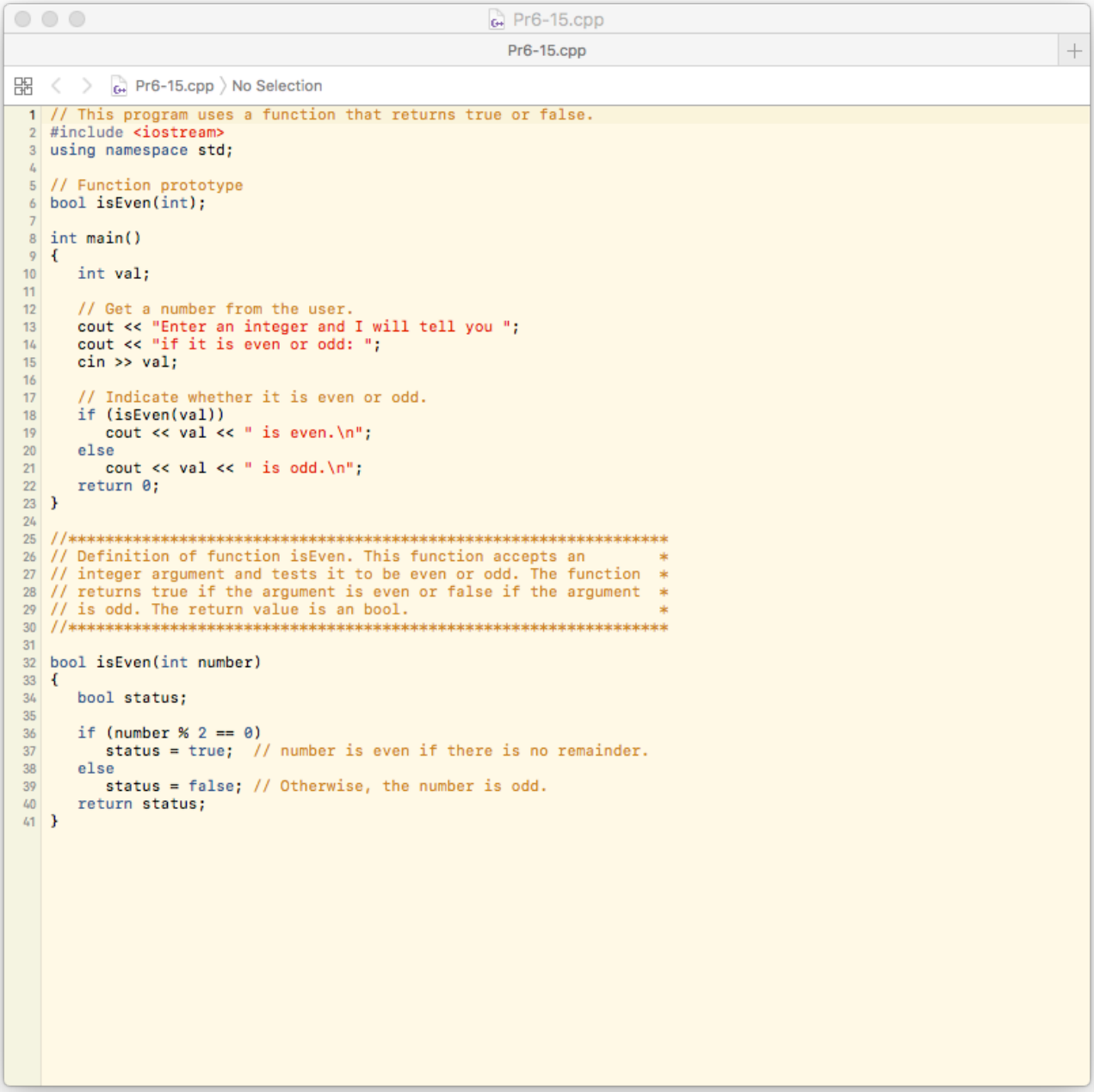
```cpp
// This program uses a function that returns a value.
#include <iostream>
using namespace std;

// Function prototype
int sum(int, int);

int main()
{
   int value1 = 20,   // The first value
       value2 = 40,   // The second value
       total;         // To hold the total

   // Call the sum function, passing the contents of
   // value1 and value2 as arguments. Assign the return
   // value to the total variable.
   total = sum(value1, value2);

   // Display the sum of the values.
   cout << "The sum of " << value1 << " and "
        << value2 << " is " << total << endl;
   return 0;
}

//*********************************************************
// Definition of function sum. This function returns  *
// the sum of its two parameters.                      *
//*********************************************************

int sum(int num1, int num2)
{
   return num1 + num2;
}
```

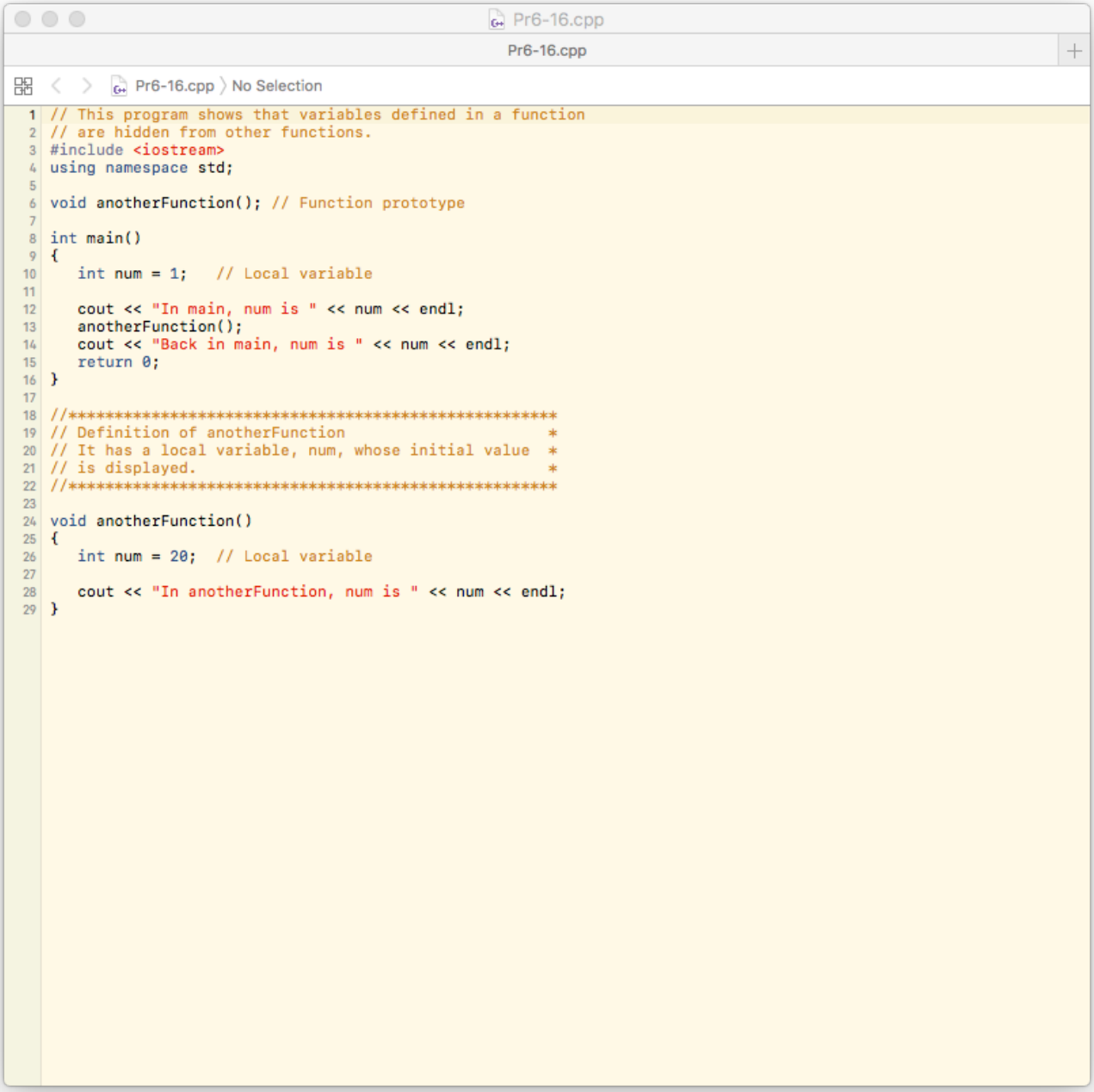



```cpp
// This program demonstrates two value-returning functions.
// The square function is called in a mathematical statement.
#include <iostream>
#include <iomanip>
using namespace std;

//Function prototypes
double getRadius();
double square(double);

int main()
{
   const double PI = 3.14159; // Constant for pi
   double radius;             // To hold the circle's radius
   double area;               // To hold the circle's area

   // Set the numeric output formatting.
   cout << fixed << showpoint << setprecision(2);

   // Get the radius of the circle.
   cout << "This program calculates the area of ";
   cout << "a circle.\n";
   radius = getRadius();

   // Caclulate the area of the circle.
   area = PI * square(radius);

   // Display the area.
   cout << "The area is " << area << endl;
   return 0;
}

//*********************************************************
// Definition of function getRadius.                      *
// This function asks the user to enter the radius of     *
// the circle and then returns that number as a double.   *
//*********************************************************

double getRadius()
{
   double rad;

   cout << "Enter the radius of the circle: ";
   cin >> rad;
   return rad;
}

//*********************************************************
// Definition of function square.                         *
// This function accepts a double argument and returns    *
// the square of the argument as a double.                *
//*********************************************************

double square(double number)
{
   return number * number;
```




```cpp
// This program converts cups to fluid ounces.
#include <iostream>
#include <iomanip>
using namespace std;

// Function prototypes
void showIntro();
double getCups();
double cupsToOunces(double);

int main()
{
   // Variables for the cups and ounces.
   double cups, ounces;

   // Set up numeric output formatting.
   cout << fixed << showpoint << setprecision(1);

   // Display an intro screen.
   showIntro();

   // Get the number of cups.
   cups = getCups();

   // Convert cups to fluid ounces.
   ounces = cupsToOunces(cups);

   // Display the number of ounces.
   cout << cups << " cups equals "
        << ounces << " ounces.\n";

   return 0;
}

//****************************************
// The showIntro function displays an      *
// introductory screen.                    *
//****************************************

void showIntro()
{
   cout << "This program converts measurements\n"
        << "in cups to fluid ounces. For your\n"
        << "reference the formula is:\n"
        << "    1 cup = 8 fluid ounces\n\n";
}

//****************************************
// The getCups function prompts the user   *
// to enter the number of cups and then    *
// returns that value as a double.         *
//****************************************

double getCups()
{
   double numCups;
```


Pr6-15.cpp
Pr6-15.cpp
Pr6-15.cpp
No Selection
// This program uses a function that returns true or false.
#include <iostream>
using namespace std;
// Function prototype
bool isEven(int);
int main()
{
int val;
// Get a number from the user.
cout << "Enter an integer and I will tell you ";
cout << "if it is even or odd: ";
cin >> val;
// Indicate whether it is even or odd.
if (isEven(val))
cout << val << " is even.\n";
else
cout << val << " is odd.\n";
return 0;
}
//***
// Definition of function isEven. This function accepts an *
// integer argument and tests it to be even or odd. The function *
// returns true if the argument is even or false if the argument *
// is odd. The return value is an bool. *
//***
bool isEven(int number)
{
bool status;
if (number % 2 == 0)
status = true; // number is even if there is no remainder.
else
status = false; // Otherwise, the number is odd.
return status;
}

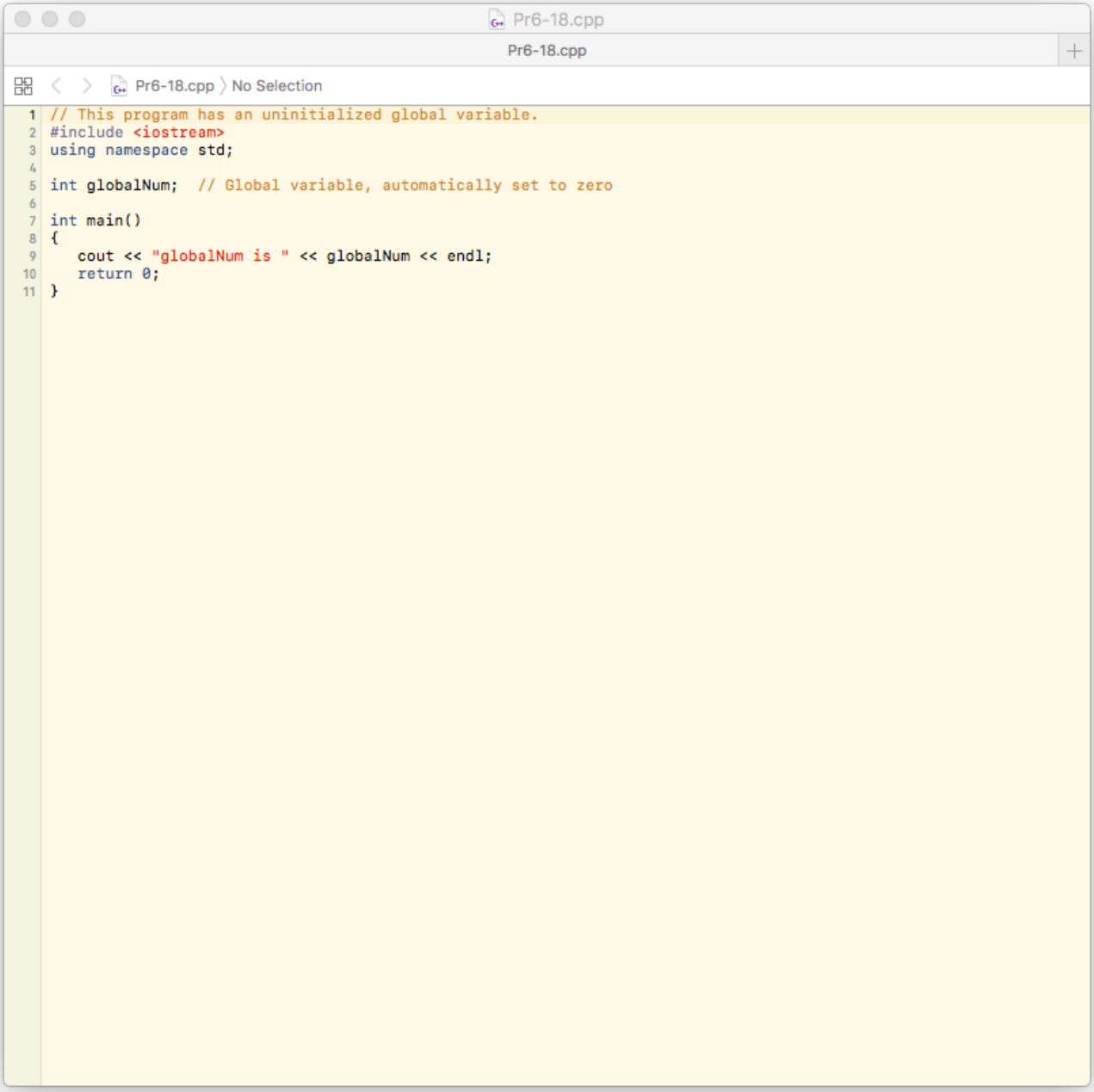

Pr6-16.cpp
Pr6-16.cpp
Pr6-16.cpp
No Selection
// This program shows that variables defined in a function
// are hidden from other functions.
#include <iostream>
using namespace std;
void anotherFunction(); // Function prototype
int main()
{
int num = 1; // Local variable
cout << "In main, num is " << num << endl;
anotherFunction();
cout << "Back in main, num is " << num << endl;
return 0;
}
//***
// Definition of anotherFunction *
// It has a local variable, num, whose initial value *
// is displayed. *
//***
void anotherFunction()
{
int num = 20; // Local variable
cout << "In anotherFunction, num is " << num << endl;
}

Pr6-17.cpp

Pr6-17.cpp

Pr6-17.cpp 〉 No Selection

```cpp
// This program shows that a global variable is visible
// to all the functions that appear in a program after
// the variable's declaration.
#include <iostream>
using namespace std;

void anotherFunction(); // Function prototype
int num = 2;            // Global variable

int main()
{
   cout << "In main, num is " << num << endl;
   anotherFunction();
   cout << "Back in main, num is " << num << endl;
   return 0;
}

//*****************************************************
// Definition of anotherFunction                      *
// This function changes the value of the             *
// global variable num.                               *
//*****************************************************

void anotherFunction()
{
   cout << "In anotherFunction, num is " << num << endl;
   num = 50;
   cout << "But, it is now changed to " << num << endl;
}
```

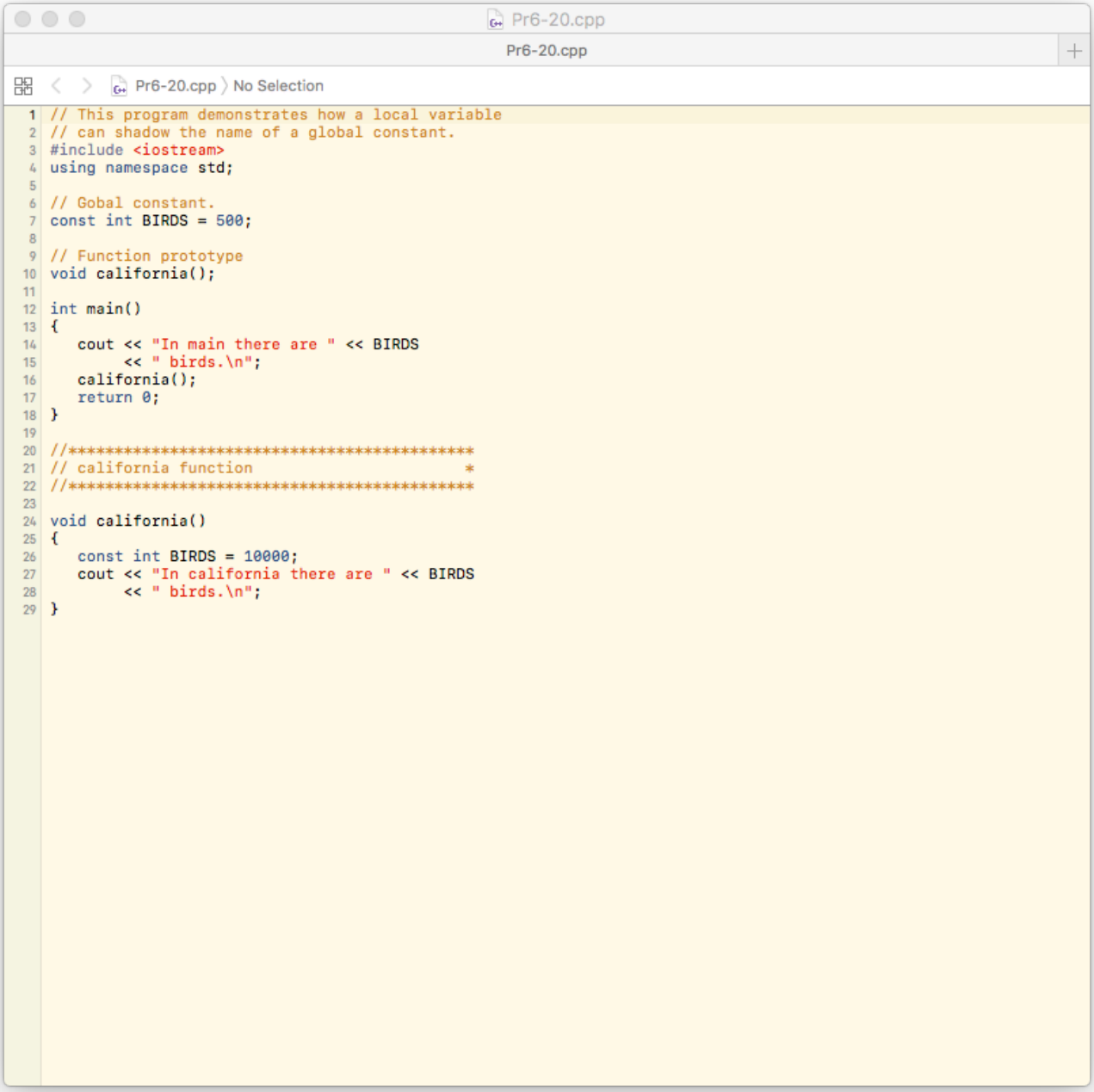

Pr6-18.cpp
Pr6-18.cpp
Pr6-18.cpp
No Selection
// This program has an uninitialized global variable.
#include <iostream>
using namespace std;
int globalNum; // Global variable, automatically set to zero
int main()
{
cout << "globalNum is " << globalNum << endl;
return 0;
}

Pr6-19.cpp

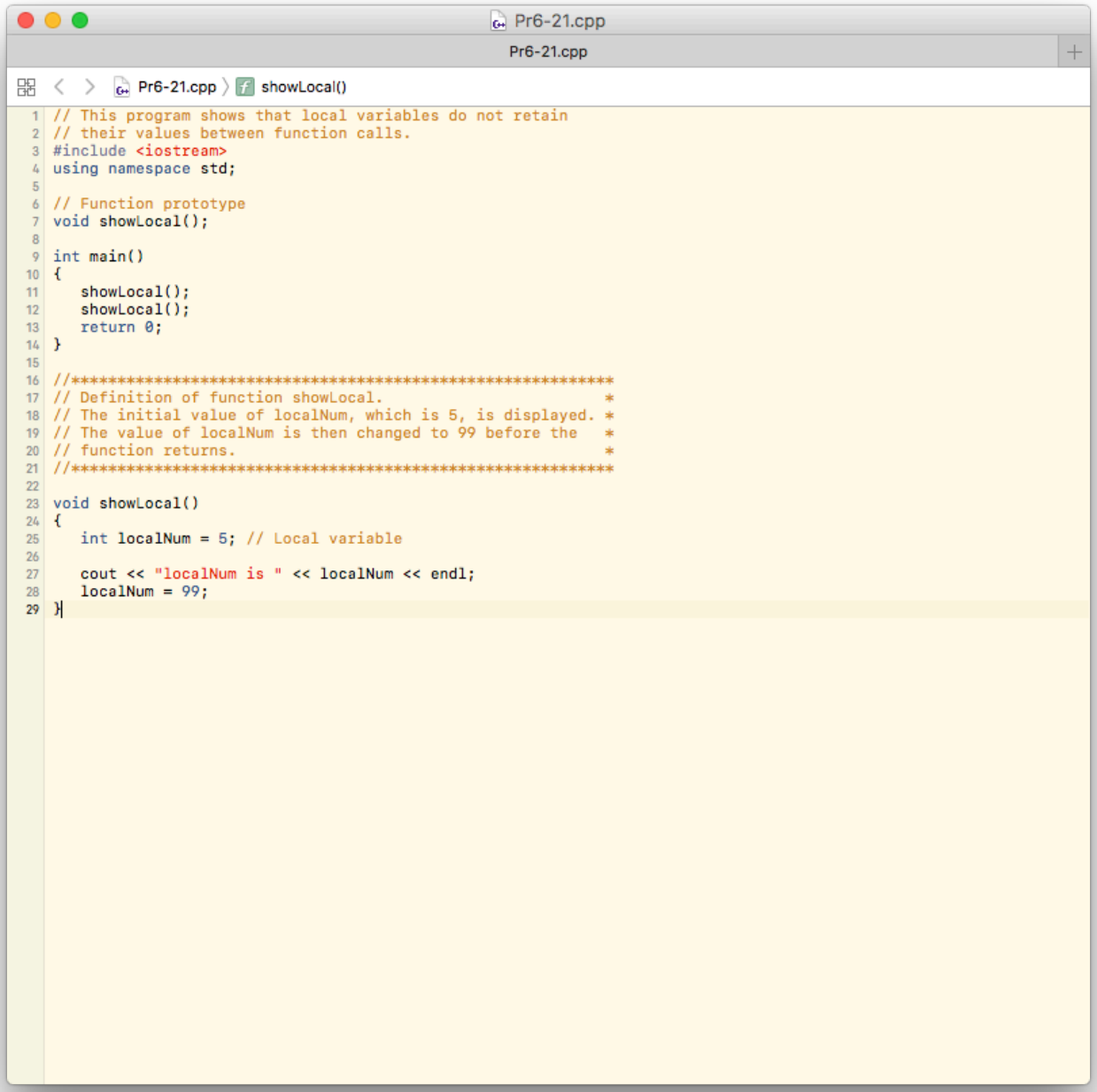
Pr6-19.cpp

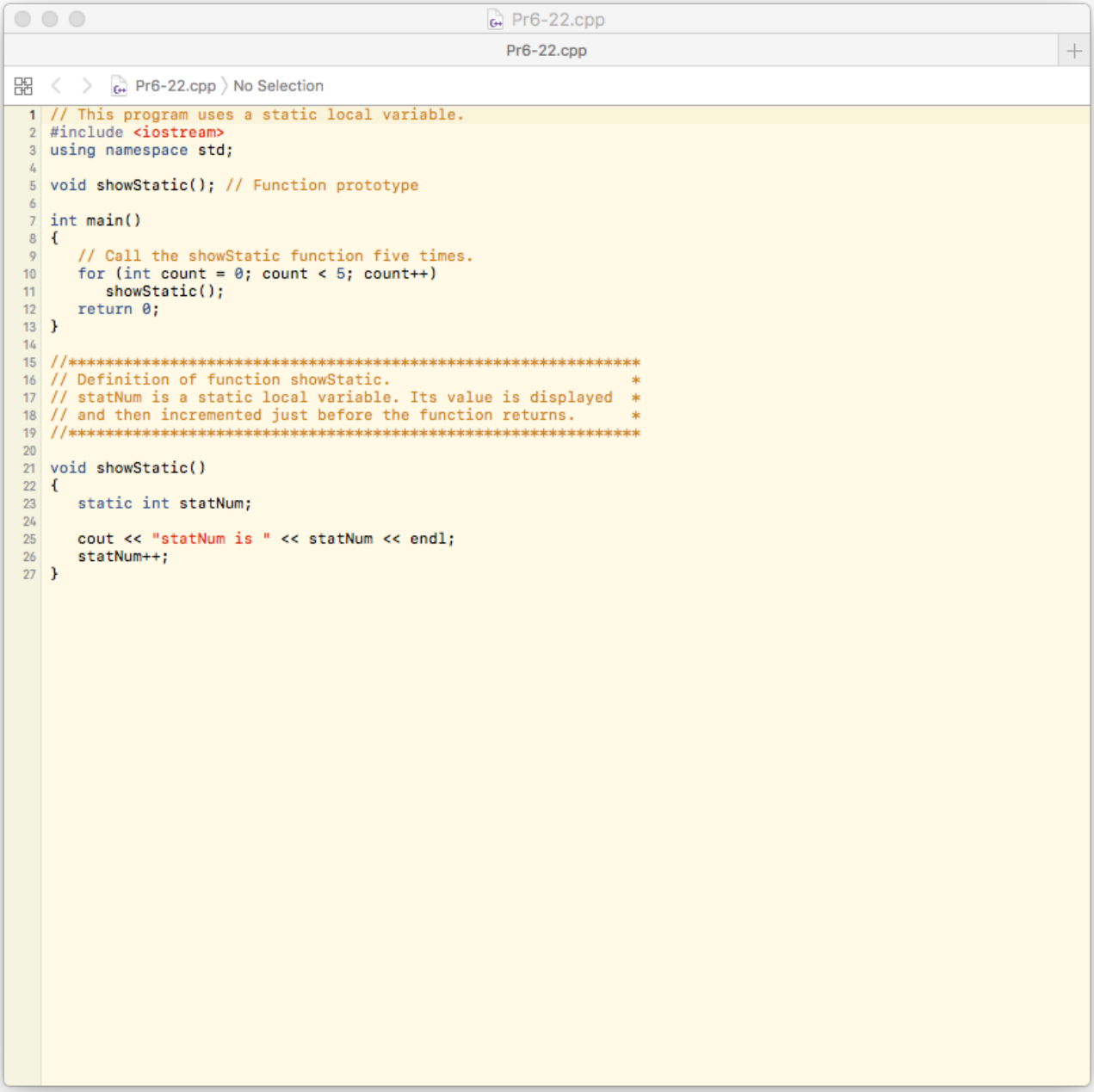
Pr6-19.cpp 〉 No Selection

```cpp
// This program calculates gross pay.
#include <iostream>
#include <iomanip>
using namespace std;

// Global constants
const double PAY_RATE = 22.55;    // Hourly pay rate
const double BASE_HOURS = 40.0;   // Max non-overtime hours
const double OT_MULTIPLIER = 1.5; // Overtime multiplier

// Function prototypes
double getBasePay(double);
double getOvertimePay(double);

int main()
{
   double hours,           // Hours worked
          basePay,         // Base pay
          overtime = 0.0,  // Overtime pay
          totalPay;        // Total pay

   // Get the number of hours worked.
   cout << "How many hours did you work? ";
   cin >> hours;

   // Get the amount of base pay.
   basePay = getBasePay(hours);

   // Get overtime pay, if any.
   if (hours > BASE_HOURS)
      overtime = getOvertimePay(hours);

   // Calculate the total pay.
   totalPay = basePay + overtime;

   // Set up numeric output formatting.
   cout << setprecision(2) << fixed << showpoint;

   // Display the pay.
   cout << "Base pay: $" << basePay << endl
        << "Overtime pay $" << overtime << endl
        << "Total pay $" << totalPay << endl;
   return 0;
}

//************************************************
// The getBasePay function accepts the number of *
// hours worked as an argument and returns the   *
// employee's pay for non-overtime hours.        *
//************************************************

double getBasePay(double hoursWorked)
{
   double basePay; // To hold base pay

   // Determine base pay
```




```cpp
// This program demonstrates how a local variable
// can shadow the name of a global constant.
#include <iostream>
using namespace std;

// Gobal constant.
const int BIRDS = 500;

// Function prototype
void california();

int main()
{
   cout << "In main there are " << BIRDS
        << " birds.\n";
   california();
   return 0;
}

//*****************************************
// california function                    *
//*****************************************

void california()
{
   const int BIRDS = 10000;
   cout << "In california there are " << BIRDS
        << " birds.\n";
}
```


Pr6-21.cpp
Pr6-21.cpp
Pr6-21.cpp
showLocal()
// This program shows that local variables do not retain
// their values between function calls.
#include <iostream>
using namespace std;
// Function prototype
void showLocal();
int main()
{
showLocal();
showLocal();
return 0;
}
//**
// Definition of function showLocal.   *
// The initial value of localNum, which is 5, is displayed. *
// The value of localNum is then changed to 99 before the   *
// function returns.   *
//**
void showLocal()
{
int localNum = 5; // Local variable
cout << "localNum is " << localNum << endl;
localNum = 99;
}




```cpp
// This program uses a static local variable.
#include <iostream>
using namespace std;

void showStatic(); // Function prototype

int main()
{
   // Call the showStatic function five times.
   for (int count = 0; count < 5; count++)
      showStatic();
   return 0;
}

//*************************************************************
// Definition of function showStatic.                         *
// statNum is a static local variable. Its value is displayed  *
// and then incremented just before the function returns.      *
//*************************************************************

void showStatic()
{
   static int statNum;

   cout << "statNum is " << statNum << endl;
   statNum++;
}
```

Pr6-23.cpp

Pr6-23.cpp

Pr6-23.cpp 〉 No Selection

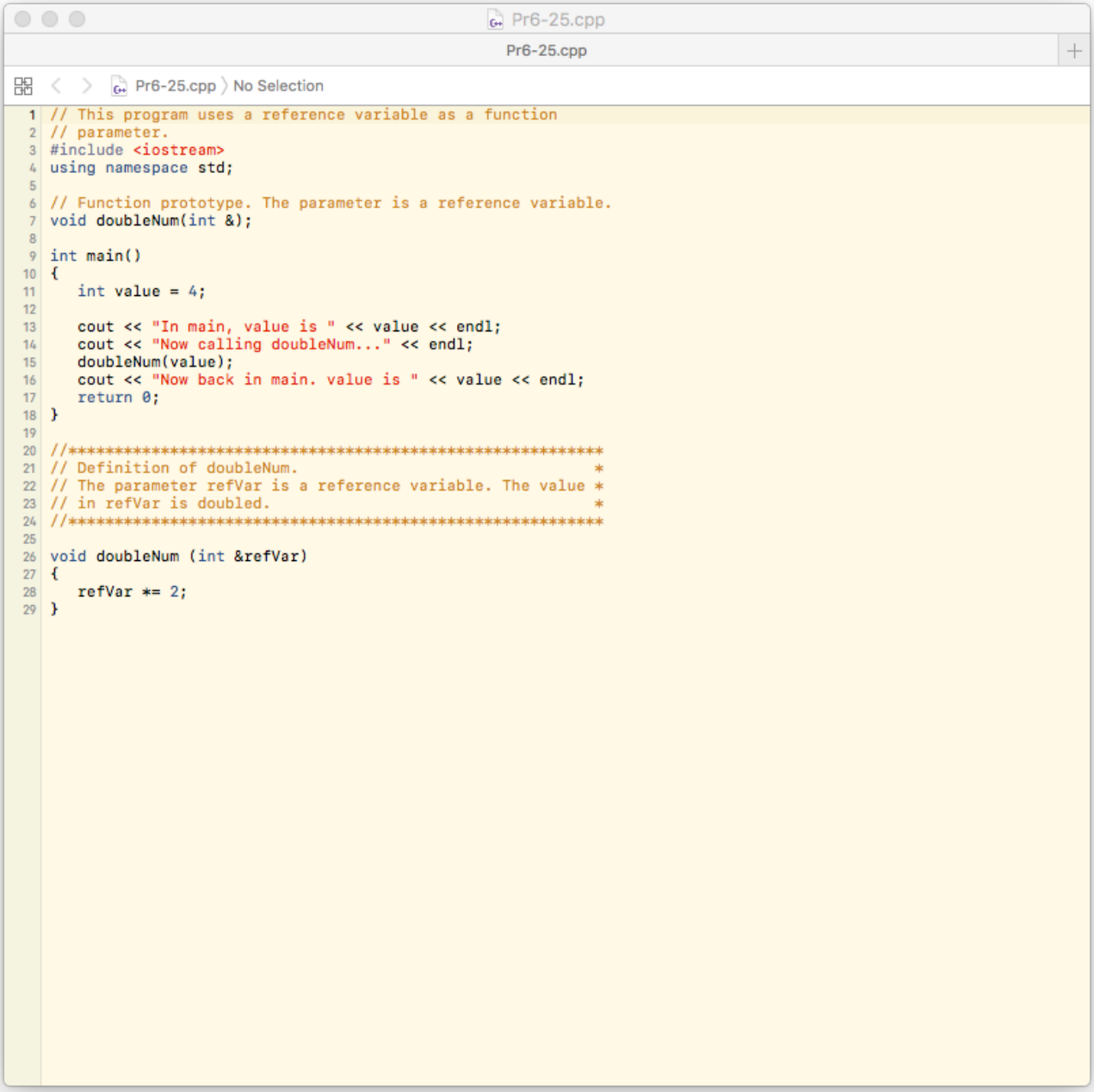
```cpp
// This program shows that a static local variable is only
// initialized once.
#include <iostream>
using namespace std;

void showStatic(); // Function prototype

int main()
{
   // Call the showStatic function five times.
   for (int count = 0; count < 5; count++)
      showStatic();
   return 0;
}

//*************************************************************
// Definition of function showStatic.                         *
// statNum is a static local variable. Its value is displayed  *
// and then incremented just before the function returns.      *
//*************************************************************

void showStatic()
{
   static int statNum = 5;

   cout << "statNum is " << statNum << endl;
   statNum++;
}
```

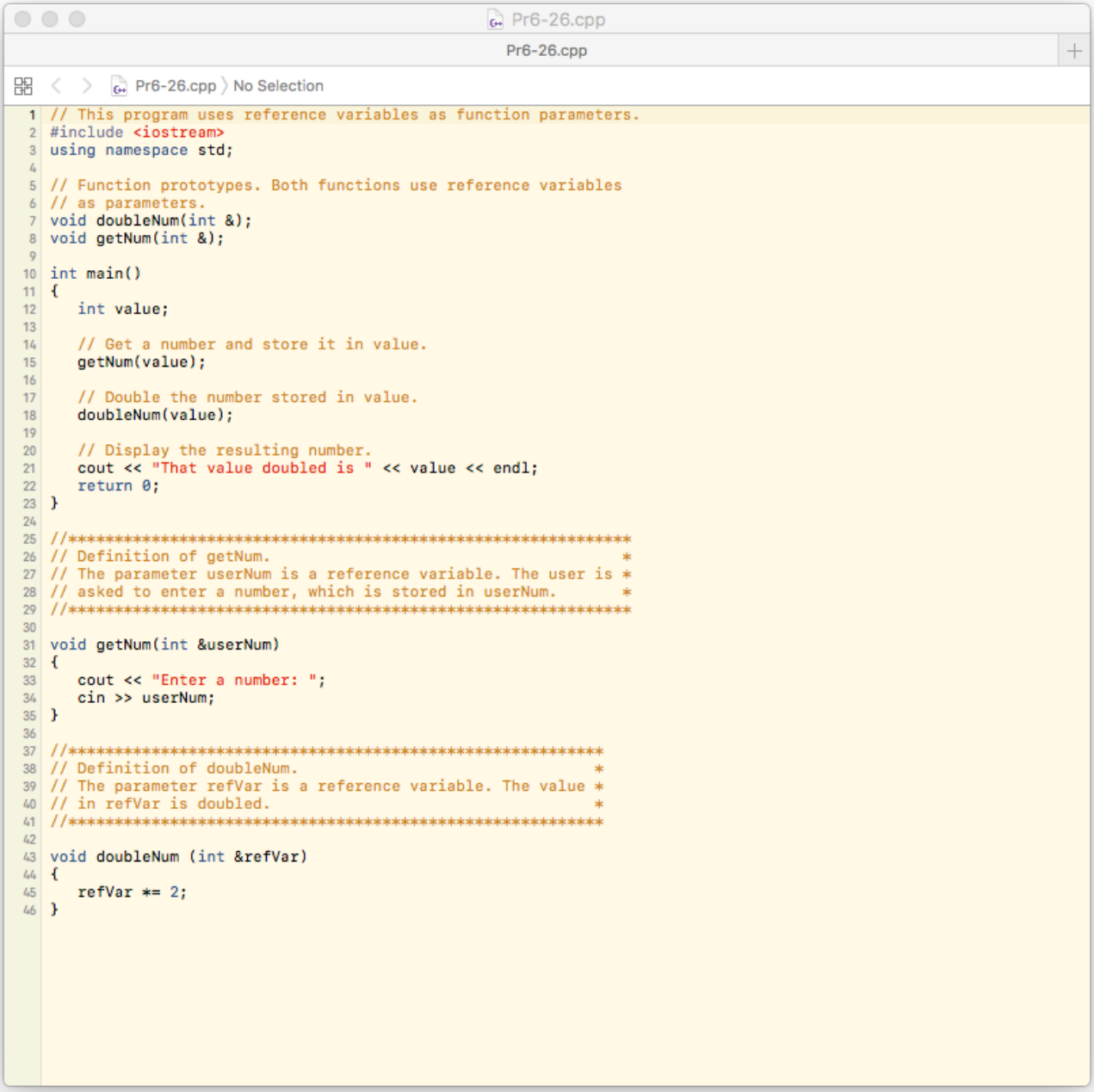
Pr6-24.cpp

Pr6-24.cpp

Pr6-24.cpp 〉 No Selection

```cpp
// This program demonstrates default function arguments.
#include <iostream>
using namespace std;

// Function prototype with default arguments
void displayStars(int = 10, int = 1);

int main()
{
   displayStars();      // Use default values for cols and rows.
   cout << endl;
   displayStars(5);     // Use default value for rows.
   cout << endl;
   displayStars(7, 3);  // Use 7 for cols and 3 for rows.
   return 0;
}

//*********************************************************
// Definition of function displayStars.                   *
// The default argument for cols is 10 and for rows is 1.*
// This function displays a square made of asterisks.     *
//*********************************************************

void displayStars(int cols, int rows)
{
   // Nested loop. The outer loop controls the rows
   // and the inner loop controls the columns.
   for (int down = 0; down < rows; down++)
   {
      for (int across = 0; across < cols; across++)
         cout << "*";
      cout << endl;
   }
}
```


Pr6-25.cpp
Pr6-25.cpp
Pr6-25.cpp
No Selection
// This program uses a reference variable as a function
// parameter.
#include <iostream>
using namespace std;
// Function prototype. The parameter is a reference variable.
void doubleNum(int &);
int main()
{
int value = 4;
cout << "In main, value is " << value << endl;
cout << "Now calling doubleNum..." << endl;
doubleNum(value);
cout << "Now back in main. value is " << value << endl;
return 0;
}
//***
// Definition of doubleNum. *
// The parameter refVar is a reference variable. The value *
// in refVar is doubled. *
//***
void doubleNum (int &refVar)
{
refVar *= 2;
}




```cpp
// This program uses reference variables as function parameters.
#include <iostream>
using namespace std;

// Function prototypes. Both functions use reference variables
// as parameters.
void doubleNum(int &);
void getNum(int &);

int main()
{
   int value;

   // Get a number and store it in value.
   getNum(value);

   // Double the number stored in value.
   doubleNum(value);

   // Display the resulting number.
   cout << "That value doubled is " << value << endl;
   return 0;
}

//*************************************************************
// Definition of getNum.                                      *
// The parameter userNum is a reference variable. The user is *
// asked to enter a number, which is stored in userNum.       *
//*************************************************************

void getNum(int &userNum)
{
   cout << "Enter a number: ";
   cin >> userNum;
}

//*************************************************************
// Definition of doubleNum.                                   *
// The parameter refVar is a reference variable. The value    *
// in refVar is doubled.                                      *
//*************************************************************

void doubleNum (int &refVar)
{
   refVar *= 2;
}
```

Pr6-27.cpp

Pr6-27.cpp

Pr6-27.cpp 〉 No Selection

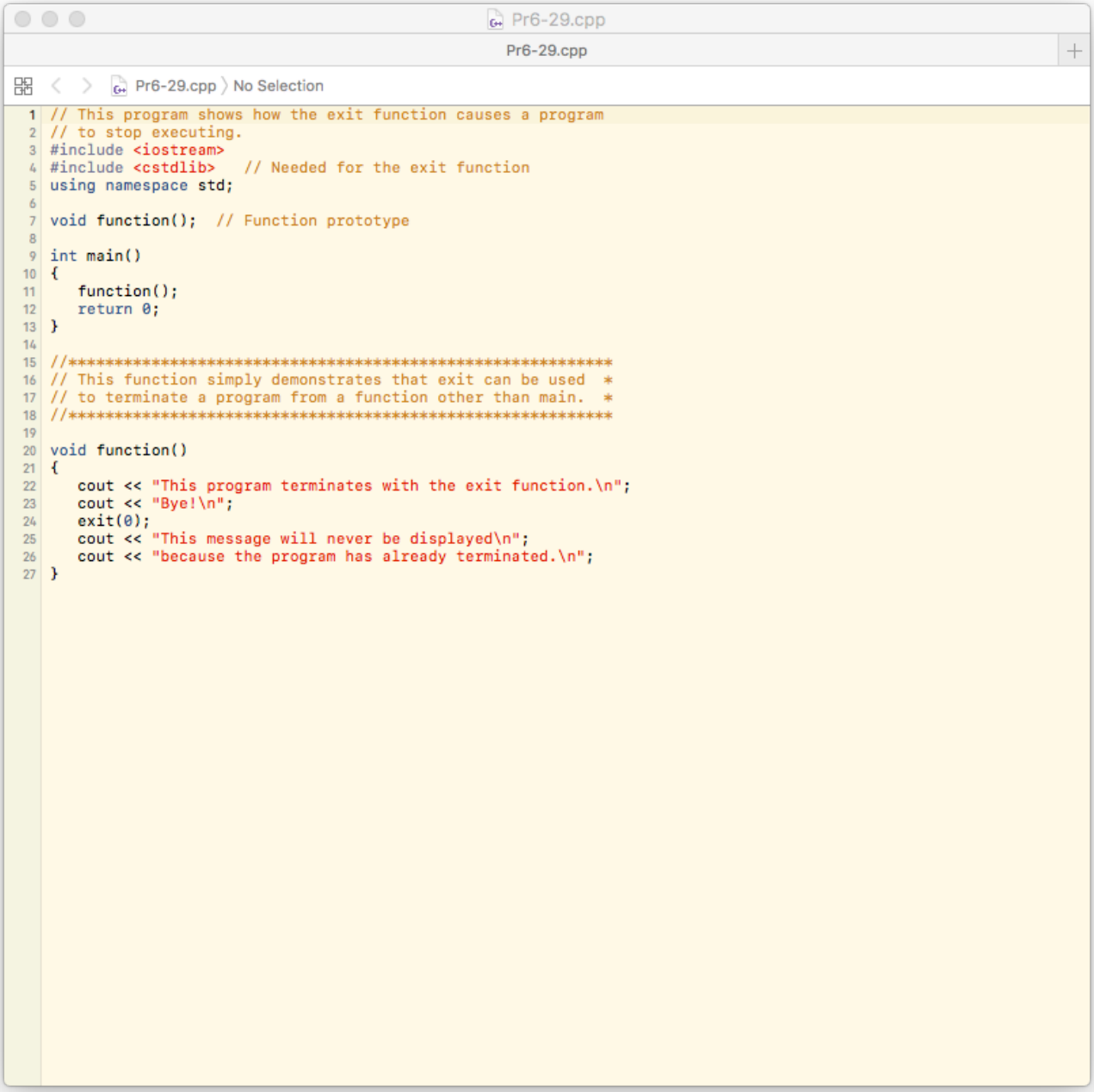
```cpp
// This program uses overloaded functions.
#include <iostream>
#include <iomanip>
using namespace std;

// Function prototypes
int square(int);
double square(double);

int main()
{
   int userInt;
   double userFloat;

   // Get an int and a double.
   cout << fixed << showpoint << setprecision(2);
   cout << "Enter an integer and a floating-point value: ";
   cin >> userInt >> userFloat;

   // Display their squares.
   cout << "Here are their squares: ";
   cout << square(userInt) << " and " << square(userFloat);
   return 0;
}

//*************************************************************
// Definition of overloaded function square.                  *
// This function uses an int parameter, number. It returns the *
// square of number as an int.                                 *
//*************************************************************

int square(int number)
{
   return number * number;
}

//*************************************************************
// Definition of overloaded function square.                  *
// This function uses a double parameter, number. It returns   *
// the square of number as a double.                           *
//*************************************************************

double square(double number)
{
   return number * number;
}
```

Pr6-28.cpp

Pr6-28.cpp

Pr6-28.cpp 〉 No Selection

```cpp
// This program demonstrates overloaded functions to calculate
// the gross weekly pay of hourly-paid or salaried employees.
#include <iostream>
#include <iomanip>
using namespace std;

// Function prototypes
void getChoice(char &);
double calcWeeklyPay(int, double);
double calcWeeklyPay(double);

int main()
{
   char selection;   // Menu selection
   int worked;       // Hours worked
   double rate;      // Hourly pay rate
   double yearly;    // Yearly salary

   // Set numeric output formatting.
   cout << fixed << showpoint << setprecision(2);

   // Display the menu and get a selection.
   cout << "Do you want to calculate the weekly pay of\n";
   cout << "(H) an hourly-paid employee, or \n";
   cout << "(S) a salaried employee?\n";
   getChoice(selection);

   // Process the menu selection.
   switch (selection)
   {
      // Hourly-paid employee
      case 'H' :
      case 'h' :  cout << "How many hours were worked? ";
                  cin >> worked;
                  cout << "What is the hour pay rate? ";
                  cin >> rate;
                  cout << "The gross weekly pay is $";
                  cout << calcWeeklyPay(worked, rate) << endl;
                  break;

      // Salaried employee
      case 'S' :
      case 's' :  cout << "What is the annual salary? ";
                  cin >> yearly;
                  cout << "The gross weekly pay is $";
                  cout << calcWeeklyPay(yearly) << endl;
                  break;
   }
   return 0;
}

//************************************************************
// Definition of function getChoice.                         *
// The parameter letter is a reference to a char.            *
// This function asks the user for an H or an S and returns  *
// the validated input.                                      *
//************************************************************

void getChoice(char &letter)
{
   // Get the user's selection.
   cout << "Enter your choice (H or S): ";
   cin >> letter;

   // Validate the selection.
   while (letter != 'H' && letter != 'h' &&
          letter != 'S' && letter != 's')
   {
      cout << "Please enter H or S: ";
      cin >> letter;
   }
}

//************************************************************
// Definition of overloaded function calcWeeklyPay.          *
// This function calculates the gross weekly pay of          *
// an hourly-paid employee. The parameter hours holds the    *
// number of hours worked. The parameter payRate holds the   *
// hourly pay rate. The function returns the weekly salary.  *
//************************************************************

double calcWeeklyPay(int hours, double payRate)
{
   return hours * payRate;
}

//************************************************************
// Definition of overloaded function calcWeeklyPay.          *
// This function calculates the gross weekly pay of          *
// a salaried employee. The parameter holds the employee's   *
// annual salary. The function returns the weekly salary.    *
//************************************************************

double calcWeeklyPay(double annSalary)
{
   return annSalary / 52;
}
```


Pr6-29.cpp
Pr6-29.cpp
Pr6-29.cpp
No Selection
// This program shows how the exit function causes a program
// to stop executing.
#include <iostream>
#include <cstdlib>   // Needed for the exit function
using namespace std;
void function(); // Function prototype
int main()
{
function();
return 0;
}
//***
// This function simply demonstrates that exit can be used *
// to terminate a program from a function other than main. *
//***
void function()
{
cout << "This program terminates with the exit function.\n";
cout << "Bye!\n";
exit(0);
cout << "This message will never be displayed\n";
cout << "because the program has already terminated.\n";
}

Pr6-30.cpp

Pr6-30.cpp

Pr6-30.cpp 〉 No Selection

```cpp
// This program is a driver for testing the showFees function.
#include <iostream>
using namespace std;

// Prototype
void showFees(double, int);

int main()
{
   // Constants for membership rates
   const double ADULT = 40.0;
   const double SENIOR = 30.0;
   const double CHILD = 20.0;

   // Perform a test for adult membership.
   cout << "Testing an adult membership...\n"
        << "Calling the showFees function with arguments "
        << ADULT << " and 10.\n";
   showFees(ADULT, 10);

   // Perform a test for senior citizen membership.
   cout << "\nTesting a senior citizen membership...\n"
        << "Calling the showFees function with arguments "
        << SENIOR << " and 10.\n";
   showFees(SENIOR, 10);

   // Perform a test for child membership.
   cout << "\nTesting a child membership...\n"
        << "Calling the showFees function with arguments "
        << CHILD << " and 10.\n";
   showFees(CHILD, 10);
   return 0;
}

//****************************************************************
// Definition of function showFees. The memberRate parameter     *
// the monthly membership rate and the months parameter holds the *
// number of months. The function displays the total charges.     *
//****************************************************************

void showFees(double memberRate, int months)
{
    cout << "The total charges are $"
         << (memberRate * months) << endl;
}
```

AreaRectangle.cpp

AreaRectangle.cpp

AreaRectangle.cpp 〉 No Selection

```cpp
// Chapter 6, Programming Challenge 2
#include <iostream>
using namespace std;

// Write the prototypes for the getLength,
// getWidth, getArea, and displayData
// functions here.

int main()
{
   double length,    // The rectangle's length
          width,     // The rectangle's width
          area;      // The rectangle's area

   // Get the rectangle's length.
   length = getLength();

   // Get the rectangle's width.
   width = getWidth();

   // Get the rectangle's area.
   are = getArea();

   // Display the rectangle's data.
   displayData(length, width, area);

   return 0;
}

//************************************************
// You must write the getLength, getWidth, getArea, *
// and displayData functions.                       *
//************************************************
```

HealthClubWithStub.cpp

HealthClubWithStub.cpp

HealthClubWithStub.cpp 〉 No Selection

```cpp
// This is a menu-driven program that makes a function call
// for each selection the user makes.
#include <iostream>
#include <iomanip>
using namespace std;

// Function prototypes
void showMenu();
void showFees(double, int);

int main()
{
   int choice;       // To hold a menu choice
   int months;       // To hold a number of months

   // Constants for membership rates
   const double ADULT = 40.0;
   const double SENIOR = 30.0;
   const double CHILD = 20.0;

   // Set up numeric output formatting.
   cout << fixed << showpoint << setprecision(2);

   do
   {
      // Display the menu and get the user's choice.
      showMenu();
      cin >> choice;

      // Validate the menu selection.
      while (choice < 1 || choice > 4)
      {
         cout << "Please enter 1, 2, 3, or 4: ";
         cin >> choice;
      }

      if (choice != 4)
      {
         // Get the number of months.
         cout << "For how many months? ";
         cin >> months;

         // Display the membership fees.
         switch (choice)
         {
            case 1:  showFees(ADULT, months);
                     break;
            case 2:  showFees(CHILD, months);
                     break;
            case 3:  showFees(SENIOR, months);
         }
      }
   } while (choice != 4);
   return 0;
}

//*********************************************************************
// Definition of function showMenu which displays the menu.       *
//*********************************************************************

void showMenu()
{
   cout << "\n\t\tHealth Club Membership Menu\n\n";
   cout << "1. Standard Adult Membership\n";
   cout << "2. Child Membership\n";
   cout << "3. Senior Citizen Membership\n";
   cout << "4. Quit the Program\n\n";
   cout << "Enter your choice: ";
}

//*********************************************************************
// showFee stub                                                   *
//*********************************************************************

void showFees(double memberRate, int months)
{
    cout << "The showFees function was called with "
         << "the following arguments:\n"
         << "memberRate: " << memberRate << endl
         << "months: " << months << endl;
}
```