Review Questions and Exercises

Short Answer
1. Why do local variables lose their values between calls to the function in which they are defined?
2. What is the difference between an argument and a parameter variable?
3. Where do you define parameter variables?
4. If you are writing a function that accepts an argument and you want to make sure the function cannot change the value of the argument, what do you do?
5. When a function accepts multiple arguments, does it matter what order the arguments are passed in?
6. How do you return a value from a function?
7. What is the advantage of breaking your application’s code into several small procedures?
8. How would a static local variable be useful?
9. Give an example where passing an argument by reference would be useful.

Fill-in-the-Blank
10. The _______ is the part of a function definition that shows the function name, return type, and parameter list.
11. If a function doesn’t return a value, the word _______ will appear as its return type.
12. Either a function’s _______ or its _______ must precede all calls to the function.
13. Values that are sent into a function are called _______.
14. Special variables that hold copies of function arguments are called _______.
15. When only a copy of an argument is passed to a function, it is said to be passed by _______.
16. A(n) _______ eliminates the need to place a function definition before all calls to the function.
17. A(n) _______ variable is defined inside a function and is not accessible outside the function.
18. _______ variables are defined outside all functions and are accessible to any function within their scope.
19. _______ variables provide an easy way to share large amounts of data among all the functions in a program.
20. Unless you explicitly initialize global variables, they are automatically initialized to _______.
21. If a function has a local variable with the same name as a global variable, only the _______ variable can be seen by the function.
22. _______ local variables retain their value between function calls.
23. The _______ statement causes a function to end immediately.
24. _______ arguments are passed to parameters automatically if no argument is provided in the function call.
25. When a function uses a mixture of parameters with and without default arguments, the parameters with default arguments must be defined _______.
26. The value of a default argument must be a(n) _______.
27. When used as parameters, _______ variables allow a function to access the parameter’s original argument.
28. Reference variables are defined like regular variables, except there is a _______ in front of the name.
29. Reference variables allow arguments to be passed by _______.
30. The _______ function causes a program to terminate.
31. Two or more functions may have the same name, as long as their _______ are different.

**Algorithm Workbench**

32. Examine the following function header, then write an example call to the function.
   ```cpp
   void showValue(int quantity)
   ```

33. The following statement calls a function named half. The half function returns a value that is half that of the argument. Write the function.
   ```cpp
   result = half(number);
   ```

34. A program contains the following function.
   ```cpp
   int cube(int num) {
      return num * num * num;
   }
   ```
   Write a statement that passes the value 4 to this function and assigns its return value to the variable result.

35. Write a function, named timesTen, that accepts an argument. When the function is called, it should display the product of its argument multiplied times 10.

36. A program contains the following function.
   ```cpp
   void display(int arg1, double arg2, char arg3) {
      cout << "Here are the values: "
           << arg1 << " " << arg2 << " "
           << arg3 << endl;
   }
   ```
   Write a statement that calls the procedure and passes the following variables to it:
   ```cpp
   int age;
   double income;
   char initial;
   ```
37. Write a function named getNumber, which uses a reference parameter variable to accept an integer argument. The function should prompt the user to enter a number in the range of 1 through 100. The input should be validated and stored in the parameter variable.

**True or False**

38. T F Functions should be given names that reflect their purpose.
39. T F Function headers are terminated with a semicolon.
40. T F Function prototypes are terminated with a semicolon.
41. T F If other functions are defined before main, the program still starts executing at function main.
42. T F When a function terminates, it always branches back to main, regardless of where it was called from.
43. T F Arguments are passed to the function parameters in the order they appear in the function call.
44. T F The scope of a parameter is limited to the function which uses it.
45. T F Changes to a function parameter always affect the original argument as well.
46. T F In a function prototype, the names of the parameter variables may be left out.
47. T F Many functions may have local variables with the same name.
48. T F Overuse of global variables can lead to problems.
49. T F Static local variables are not destroyed when a function returns.
50. T F All static local variables are initialized to −1 by default.
51. T F Initialization of static local variables only happens once, regardless of how many times the function in which they are defined is called.
52. T F When a function with default arguments is called and an argument is left out, all arguments that come after it must be left out as well.
53. T F It is not possible for a function to have some parameters with default arguments and some without.
54. T F The exit function can only be called from main.
55. T F A stub is a dummy function that is called instead of the actual function it represents.

**Find the Errors**

Each of the following functions has errors. Locate as many errors as you can.

56. ```
void total(int value1, value2, value3)
{
    return value1 + value2 + value3;
}
```

57. ```
float average(int value1, int value2, int value3)
{
    float average;
    average = value1 + value2 + value3 / 3;
}
```
58. void area(int length = 30, int width)
    {
        return length * width;
    }

59. void getValue(int value&)
    {
        cout << "Enter a value: ";
        cin >> value&;
    }

60. (Overloaded functions)
    int getValue()
    {
        int inputValue;
        cout << "Enter an integer: ";
        cin >> inputValue;
        return inputValue;
    }
    double getValue()
    {
        double inputValue;
        cout << "Enter a floating-point number: ";
        cin >> inputValue;
        return inputValue;
    }

**Programming Challenges**

1. Markup

Write a program that asks the user to enter an item’s wholesale cost and its markup percentage. It should then display the item’s retail price. For example:

- If an item’s wholesale cost is 5.00 and its markup percentage is 100%, then the item’s retail price is 10.00.
- If an item’s wholesale cost is 5.00 and its markup percentage is 50%, then the item’s retail price is 7.50.

The program should have a function named calculateRetail that receives the wholesale cost and the markup percentage as arguments, and returns the retail price of the item.

*Input Validation: Do not accept negative values for either the wholesale cost of the item or the percent markup.*

2. Rectangle Area—Complete the Program

The Student CD contains a partially written program named AreaRectangle.cpp. Your job is to complete the program. When it is complete, the program will ask the user to enter the width and length of a rectangle, and then display the rectangle’s area. The program calls the following functions, which have not been written:

- **getLength** – This function should ask the user to enter the rectangle’s length, and then return that value as a double.
• getLength – This function should ask the user to enter the rectangle's length, and then return that value as a double.
• getArea – This function should accept the rectangle's length and width as arguments, and return the rectangle's area. The area is calculated by multiplying the length by the width.
• displayData – This function should accept the rectangle's length, width, and area as arguments, and display them in an appropriate message on the screen.

3. Winning Division

Write a program that determines which of a company's four divisions (Northeast, Southeast, Northwest, and Southwest) had the greatest sales for a quarter. It should include the following two functions, which are called by main.

• double getSales() is passed the name of a division. It asks the user for a division's quarterly sales figure, validates the input, then returns it. It should be called once for each division.
• void findHighest() is passed the four sales totals. It determines which is the largest and prints the name of the high grossing division, along with its sales figure.

*Input Validation: Do not accept dollar amounts less than $0.00.*

4. Safest Driving Area

Write a program that determines which of 5 geographic regions within a major city (north, south, east, west, and central) had the fewest reported automobile accidents last year. It should have the following two functions, which are called by main.

• int getNumAccidents() is passed the name of a region. It asks the user for the number of automobile accidents reported in that region during the last year, validates the input, then returns it. It should be called once for each city region.
• void findLowest() is passed the five accident totals. It determines which is the smallest and prints the name of the region, along with its accident figure.

*Input Validation: Do not accept an accident number that is less than 0.*

5. Falling Distance

When an object is falling because of gravity, the following formula can be used to determine the distance the object falls in a specific time period:

\[ d = \frac{1}{2} gt^2 \]

The variables in the formula are as follows: \( d \) is the distance in meters, \( g \) is 9.8, and \( t \) is the amount of time, in seconds, that the object has been falling.

Write a function named fallingDistance that accepts an object's falling time (in seconds) as an argument. The function should return the distance, in meters, that the object has fallen during that time interval. Write a program that demonstrates the function by calling it in a loop that passes the values 1 through 10 as arguments, and displays the return value.
6. Kinetic Energy

In physics, an object that is in motion is said to have kinetic energy. The following formula can be used to determine a moving object’s kinetic energy:

\[ KE = \frac{1}{2} mv^2 \]

The variables in the formula are as follows: \( KE \) is the kinetic energy, \( m \) is the object’s mass in kilograms, and \( v \) is the object’s velocity, in meters per second.

Write a function named `kineticEnergy` that accepts an object’s mass (in kilograms) and velocity (in meters per second) as arguments. The function should return the amount of kinetic energy that the object has. Demonstrate the function by calling it in a program that asks the user to enter values for mass and velocity.

7. Celsius Temperature Table

The formula for converting a temperature from Fahrenheit to Celsius is

\[ C = \frac{5}{9}(F - 32) \]

where \( F \) is the Fahrenheit temperature and \( C \) is the Celsius temperature. Write a function named `celsius` that accepts a Fahrenheit temperature as an argument. The function should return the temperature, converted to Celsius. Demonstrate the function by calling it in a loop that displays a table of the Fahrenheit temperatures 0 through 20 and their Celsius equivalents.

8. Lowest Score Drop

Write a program that calculates the average of a group of test scores, where the lowest score in the group is dropped. It should use the following functions:

- `void getScore()` should ask the user for a test score, store it in a reference parameter variable, and validate it. This function should be called by `main` once for each of the five scores to be entered.
- `void calcAverage()` should calculate and display the average of the four highest scores. This function should be called just once by `main`, and should be passed the five scores.
- `int findLowest()` should find and return the lowest of the five scores passed to it. It should be called by `calcAverage`, who uses the function to determine which of the five scores to drop.

*Input Validation: Do not accept test scores lower than 0 or higher than 100.*

9. Star Search

A particular talent competition has 5 judges, each of whom awards a score between 0 and 10 to each performer. Fractional scores, such as 8.3, are allowed. A performer’s final score is determined by dropping the highest and lowest score received, then averaging the 3 remaining scores. Write a program that uses this method to calculate a contestant’s score. It should include the following functions:

- `void getJudgeData()` should ask the user for a judge’s score, store it in a reference parameter variable, and validate it. This function should be called by `main` once for each of the 5 judges.
void calcScore() should calculate and display the average of the 3 scores that remain after dropping the highest and lowest scores the performer received. This function should be called just once by main, and should be passed the 5 scores.

The last two functions, described below, should be called by calcScore, who uses the returned information to determine which of the scores to drop.

- int findLowest() should find and return the lowest of the 5 scores passed to it.
- int findHighest() should find and return the highest of the 5 scores passed to it.

Input Validation: Do not accept judge scores lower than 0 or higher than 10.

10. Days Out

Write a program that calculates the average number of days a company's employees are absent. The program should have the following functions:

- A function called by main that asks the user for the number of employees in the company. This value should be returned as an int. (The function accepts no arguments.)
- A function called by main that accepts one argument: the number of employees in the company. The function should ask the user to enter the number of days each employee missed during the past year. The total of these days should be returned as an int.
- A function called by main that takes two arguments: the number of employees in the company and the total number of days absent for all employees during the year. The function should return, as a double, the average number of days absent. (This function does not perform screen output and does not ask the user for input.)

Input Validation: Do not accept a number less than 1 or for the number of employees. Do not accept a negative number for the days any employee missed.

11. Order Status

The Middletown Wholesale Copper Wire Company sells spools of copper wiring for $100 each. Write a program that displays the status of an order. The program should have a function that asks for the following data:

- The number of spools ordered.
- The number of spools in stock.
- If there are special shipping and handling charges.

(Shipping and handling is normally $10 per spool.) If there are special charges, it should ask for the special charges per spool.

The gathered data should be passed as arguments to another function that displays:

- The number of spools ready to ship from current stock.
- The number of spools on backorder (if the number ordered is greater than what is in stock).
- Subtotal of the portion ready to ship (the number of spools ready to ship times $100).
- Total shipping and handling charges on the portion ready to ship.
- Total of the order ready to ship.

The shipping and handling parameter in the second function should have the default argument 10.00.
Input Validation: Do not accept numbers less than 1 for spools ordered. Do not accept a number less than 0 for spools in stock or shipping and handling charges.

12. Overloaded Hospital

Write a program that computes and displays the charges for a patient's hospital stay. First, the program should ask if the patient was admitted as an in-patient or an out-patient. If the patient was an in-patient, the following data should be entered:

- The number of days spent in the hospital
- The daily rate
- Hospital medication charges
- Charges for hospital services (lab tests, etc.)

The program should ask for the following data if the patient was an out-patient:

- Charges for hospital services (lab tests, etc.)
- Hospital medication charges

The program should use two overloaded functions to calculate the total charges. One of the functions should accept arguments for the in-patient data, while the other function accepts arguments for out-patient information. Both functions should return the total charges.

Input Validation: Do not accept negative numbers for any data.

13. Population

In a population, the birth rate is the percentage increase of the population due to births and the death rate is the percentage decrease of the population due to deaths. Write a program that displays the size of a population for any number of years. The program should ask for the following data:

- The starting size of a population
- The annual birth rate
- The annual death rate
- The number of years to display

Write a function that calculates the size of the population for a year. The formula is

\[ N = P + BP - DP \]

Where \( N \) is the new population size, \( P \) is the previous population size, \( B \) is the birth rate, and \( D \) is the death rate.

Input Validation: Do not accept numbers less than 2 for the starting size. Do not accept negative numbers for birth rate or death rate. Do not accept numbers less than 1 for the number of years.

14. Transient Population

Modify Programming Challenge 13 to also consider the effect on population caused by people moving into or out of a geographic area. Given as input a starting population size, the annual birth rate, the annual death rate, the number of individuals typically move into the area each year, and the number of individuals typically leave the area each year, the program should project what the population will be.
from now. You can either prompt the user to input a value for numyears, or you can set it within the program.

*Input Validation: Do not accept numbers less than 2 for the starting size. Do not accept negative numbers for birth rate, death rate, arrivals, or departures.*

15. Paint Job Estimator

A painting company has determined that for every 115 square feet of wall space, one gallon of paint and eight hours of labor will be required. The company charges $18.00 per hour for labor. Write a modular program that allows the user to enter the number of rooms that are to be painted and the price of the paint per gallon. It should also ask for the square feet of wall space in each room. It should then display the following data:

- The number of gallons of paint required
- The hours of labor required
- The cost of the paint
- The labor charges
- The total cost of the paint job

*Input validation: Do not accept a value less than 1 for the number of rooms. Do not accept a value less than $10.00 for the price of paint. Do not accept a negative value for square footage of wall space.*

16. Using Files—Hospital Report

Modify Programming Challenge 12, Overloaded Hospital, to write the report it creates to a file. Print the contents of the file to hand in to your instructor.

17. Stock Profit

The profit from the sale of a stock can be calculated as follows:

\[
\text{Profit} = (\text{NS} \times \text{SP}) - \text{SC} - (\text{NS} \times \text{PP}) + \text{PC}
\]

where NS is the number of shares, SP is the sale price per share, SC is the sale commission paid, PP is the purchase price per share, and PC is the purchase commission paid. If the calculation yields a positive value, then the sale of the stock resulted in a profit. If the calculation yields a negative number, then the sale resulted in a loss.

Write a function that accepts as arguments the number of shares, the purchase price per share, the purchase commission paid, the sale price per share, and the sale commission paid. The function should return the profit (or loss) from the sale of stock.

Demonstrate the function in a program that asks the user to enter the necessary data and displays the amount of the profit or loss.

18. Multiple Stock Sales

Use the function that you wrote for Programming Challenge 17 (Stock Profit) in a program that calculates the total profit or loss from the sale of multiple stocks. The program should ask the user for the number of stock sales, and the necessary data for each stock sale. It should accumulate the profit or loss for each stock sale and then display the total.
19. isPrime Function

A prime number is a number that is only evenly divisible by itself and 1. For example, the number 5 is prime because it can only be evenly divided by 1 and 5. The number 6, however, is not prime because it can be divided evenly by 1, 2, 3, and 6.

Write a function name isPrime, which takes an integer as an argument and returns true if the argument is a prime number, or false otherwise. Demonstrate the function in a complete program.

**TIP:** Recall that the % operator divides one number by another, and returns the remainder of the division. In an expression such as num1 % num2, the % operator will return 0 if num1 is evenly divisible by num2.

20. Prime Number List

Use the isPrime function that you wrote in Programming Challenge 19 in a program that stores a list of all the prime numbers from 1 through 100 in a file.

**Group Project**

21. Travel Expenses

This program should be designed and written by a team of students. Here are some suggestions:

- One student should design function main, which will call the other functions in the program. The remainder of the functions will be designed by other members of the team.
- The requirements of the program should be analyzed so each student is given about the same work load.
- The parameters and return types of each function should be decided in advance.
- Stubs and drivers should be used to test and debug the program.
- The program can be implemented either as a multi-file program, or all the functions can be cut and pasted into the main file.

Here is the assignment: Write a program that calculates and displays the total travel expenses of a businessperson on a trip. The program should have functions that ask for and return the following:

- The total number of days spent on the trip
- The time of departure on the first day of the trip, and the time of arrival back home on the last day of the trip
- The amount of any round-trip airfare
- The amount of any car rentals
- Miles driven, if a private vehicle was used. Calculate the vehicle expense as $0.27 per mile driven
- Parking fees (The company allows up to $6 per day. Anything in excess of this must be paid by the employee.)
Review Questions and Exercises

- Taxi fees, if a taxi was used anytime during the trip (The company allows up to $10 per day, for each day a taxi was used. Anything in excess of this must be paid by the employee.)
- Conference or seminar registration fees
- Hotel expenses (The company allows up to $90 per night for lodging. Anything in excess of this must be paid by the employee.)
- The amount of each meal eaten. On the first day of the trip, breakfast is allowed as an expense if the time of departure is before 7 a.m. Lunch is allowed if the time of departure is before 12 noon. Dinner is allowed on the first day if the time of departure is before 6 p.m. On the last day of the trip, breakfast is allowed if the time of arrival is after 8 a.m. Lunch is allowed if the time of arrival is after 1 p.m. Dinner is allowed on the last day if the time of arrival is after 7 p.m. The program should only ask for the amounts of allowable meals. (The company allows up to $9 for breakfast, $12 for lunch, and $16 for dinner. Anything in excess of this must be paid by the employee.)

The program should calculate and display the total expenses incurred by the businessperson, the total allowable expenses for the trip, the excess that must be reimbursed by the businessperson, if any, and the amount saved by the businessperson if the expenses were under the total allowed.

Input Validation: Do not accept negative numbers for any dollar amount or for miles driven in a private vehicle. Do not accept numbers less than 1 for the number of days. Only accept valid times for the time of departure and the time of arrival.

Serendipity Booksellers Software Development Project—Part 6: A Problem-Solving Exercise

1. Function Name Change

It is now time to make one program from the separate files you have created. Perform the following function name changes:

- Change the name of function main in cashier.cpp to cashier.
- Change the name of function main invmenu.cpp to invmenu.
- Change the name of function main bookinfo.cpp to bookinfo.
- Change the name of function main in reports.cpp to reports.

Save each file after you have changed the name of its function main.

2. Development Strategy

You must now decide if you are going to develop the project as a multi-file program or simply merge all the functions listed above into the mainmenu.cpp file. (See Appendix J, Multisource file programs)

Multi-File Program

If you decide on the multi-file program approach, follow the directions in your compiler manuals to create a project or "make files" for this program. The files that are part of the project are mainmenu.cpp, cashier.cpp, invmenu.cpp, bookinfo.cpp, and reports.cpp. mainmenu.cpp will be the main file.