CHAPTER SUMMARY

General Topics

Program Routines
Value-Returning vs. Non-Value-Returning Functions
Side Effects (of Function Calls)
Parameter Passing: Actual Arguments vs. Formal Parameters
Local Scope and Local Variables
Global Scope and Global Variables
Variable Lifetime

Python-Specific Programming Topics

Defining Functions in Python
Built-in Functions of Python
Value-Returning and Non-Value-Returning Functions in Python
Tuple Assignment in Python
Mutable vs. Immutable Arguments in Python
Local vs. Global Variables in Python

CHAPTER EXERCISES

Section 5.1

1. Function `avg` returns the average of three values, as given in the chapter. Which of the following statements, each making calls to function `avg`, are valid? (Assume that all variables are of numeric type.)
   (a) `result = avg(n1, n2)`
   (b) `result = avg(n1, n2, avg(n3, n4, n5))`
   (c) `result = avg(n1 + n2, n3 + n4, n5 + n6)`
   (d) `print(avg(n1, n2, n3))`
   (e) `avg(n1, n2, n3)`
2. Which of the following statements, each involving calls to function `displayWelcome` displaying a welcome message on the screen as given in the chapter, are valid?
   (a) `print(displayWelcome)`
   (b) `displayWelcome`
   (c) `result = displayWelcome()`
   (d) `displayWelcome()`

Section 5.2

3. Suppose there are nine variables, each holding an integer value as shown below, for which the average of the largest value in each line of variables is to be computed.

```
num1 = 10      num2 = 20      num3 = 25      max1 = 25
num4 = 5       num5 = 15      num6 = 35      max2 = 35
num7 = 20      num8 = 30      num9 = 25      max3 = 30

average = (max1 + max2 + max3) / 3.0
        = (25 + 35 + 30) / 3.0
        = 30.0
```

   Using functions `avg` and `max`, give an expression that computes the average as shown above.
4. Assume that there exists a Boolean function named `isLeapYear` that determines if a given year is a leap year or not. Give an appropriate if statement that prints "Year is a Leap Year" if the year passed is a leap year, and "Year is Not a Leap Year" otherwise, for variable `year`.

5. For the following function definition and associated function calls,

```
def somefunction(n1, n2):
   .
   .
# main
num1 = 10
somefunction(num1, 15)
```

(a) List all the formal parameters.
(b) List all the actual arguments.

6. For the following function, indicate whether each function call is proper or not. If improper, explain why.

```
def gcd(n1, n2):
```

function gcd calculates the greatest common divisor of n1 and n2, with the requirement that n1 be less than or equal to n2, and n1 and n2 are integer values.

(a)
```
a = 10
b = 20
result = gcd(a, b)
```
(b)
```
a = 10.0
b = 20
result = gcd(a, b)
```
(c)
```
a = 20
b = 10
result = gcd(b, a)
```
(d)
```
a = 10
b = 20
c = 30
result = gcd(gcd(a, b), c)
```
(e)
```
a = 10
b = 20
c = 30
print(gcd(a, gcd(c, b)))
```

PYTHON PROGRAMMING EXERCISES

P1. Write a Python function named `zeroCheck` that is given three integers, and returns true if any of the integers is 0, otherwise it returns false.

P2. Write a Python function named `ordered3` that is passed three integers, and returns true if the three integers are in order from smallest to largest, otherwise it returns false.

P3. Write a Python function named `modCount` that is given a positive integer, `n`, and a second positive integer, `m <= n`, and returns how many numbers between `1` and `n` are evenly divisible by `m`.

P4. Write a Python function named `helloWorld` that displays "`Hello World, my name is` *name*", for any given name passed to the routine.

P5. Write a Python function named `printAsterisks` that is passed a positive integer value `n`, and prints out a line of `n` asterisks. If `n` is greater than 75, then only 75 asterisks should be displayed.

P6. Write a Python function named `getContinue` that displays to the user "`Do you want to continue (y/n): `", and continues to prompt the user until either uppercase or lowercase `'y'` or `'n'` is entered, returning (lowercase) `'y'` or `'n'` as the function value.

P7. Implement a Python function that is passed a list of numeric values and a particular threshold value, and returns the list with all values above the given threshold value set to 0. The list should be altered as a side effect to the function call, and not by function return value.

P8. Implement the Python function described in question P7 so that the altered list is returned as a function value, rather than by side effect.

PROGRAM MODIFICATION PROBLEMS

M1. Temperature Conversion Program: Adding Kelvin Scale
Modify the Temperature Conversion program in section 5.1.3 so that it allows the user to select temperature conversion to include degrees Kelvin, in addition to degrees Fahrenheit and degrees Celsius. Include input error checking for inappropriate temperature values. (NOTE: Refer to questions M1 and M2 from Chapter 3.)

M2. GPA Calculation Program: Accommodating First-Semester Students
Modify the GPA Calculation program in section 5.2.7 so that it asks the student if this is their first semester. If so, the program should only prompt for their current semester grades, and not their cumulative GPA and total earned credits, and display their semester GPA and cumulative GPA accordingly.

M3. GPA Calculation Program: Allowing for Plus/Minus Grading
Modify the GPA Calculation program in section 5.2.7 so that it is capable of calculating a GPA for plus/minus letter grades: A, A−, B+, B, B−, and so forth.

M4. Credit Card Calculation Program: Summarized Output
Modify the Credit Card Calculation program in section 5.3 so that the user is given the option of either displaying the balance and interest paid month-by-month as currently written, or to simply have the total number of months and the total interest paid without the month-by-month details.

M5. Credit Card Calculation Program: Adjustable Minimum Payment
Modify the Credit Card Calculation program in section 5.3 so that the user can enter the percentage from which the minimum monthly payment is calculated. Also modify the program so that this minimum payment percentage is displayed along with the other credit card related information.

M6. Credit Card Calculation Program: Recalculation with New Balance
Modify the Credit Card Calculation program in section 5.3 so that the program will allow the user to recalculate a new payoff schedule for a new entered balance.

PROGRAM DEVELOPMENT PROBLEMS

D1. Metric Conversion Program
Develop and test a Python program that allows the user to convert between the metric measurements of millimeter, centimeter, meter, kilometer, and inches, feet, yards, and miles. The program should be written so that any one measurement can be converted to the other.

D2. GPA Projection Program
Develop and test a Python program that lets the user enter their current cumulative GPA, their total credits earned, and the number of credits they are currently taking. The program should then request from the