**Review Questions**

**Multiple Choice**

1. This is a prewritten function that is built into a programming language.

a. standard function

b. library function

c. custom function

d. cafeteria function

2. This term describes any mechanism that accepts input, performs some operation that

cannot be seen on the input, and produces output.

a. glass box

b. white box

c. opaque box

d. black box

3. This standard library function returns a random integer within a specified range of values.

a. random

b. randint

c. random\_integer

d. uniform

4. This standard library function returns a random floating-point number in the range of

0.0 up to 1.0 (but not including 1.0).

a. random

b. randint

c. random\_integer

d. uniform

5. This standard library function returns a random floating-point number within a specified

range of values.

a. random

b. randint

c. random\_integer

d. uniform

6. This statement causes a function to end and sends a value back to the part of the program

that called the function.

a. end

b. send

c. exit

d. return

7. This is a design tool that describes the input, processing, and output of a function.

a. hierarchy chart

b. IPO chart

c. datagram chart

d. data processing chart

8. This type of function returns either True or False.

a. Binary

b. true\_false

c. Boolean

d. logical

9. This is math module function.

a. derivative

b. factor

c. sqrt

d. differentiate

10. A menu is a \_\_\_\_\_\_\_\_\_\_.

a. case structure that selects an operation in a program

b. group of modules that perform individual tasks

c. list of operations displayed on the screen that the user may choose from

d. table of Boolean choices

**True or False**

1. Some library functions are built into the Python interpreter.

2. You do not have to have an import statement in a program to use the functions in the

random module.

3. Complex mathematical expressions can sometimes be simplified by breaking out part

of the expression and putting it in a function.

4. A function in Python can return more than one value.

5. IPO charts provide only brief descriptions of a function’s input, processing, and output,

but do not show the specific steps taken in a function.

**Short Answer**

1. Suppose you want to select a random number from the following sequence:

0, 5, 10, 15, 20, 25, 30

What library function would you use?

2. What statement do you have to have in a value-returning function?

3. What three things are listed on an IPO chart?

4. What is a Boolean function?

5. What are the advantages of breaking a large program into modules?

**Algorithm Workbench**

1. Write a statement that generates a random number in the range of 1 through 100 and

assigns it to a variable named rand.

2. The following statement calls a function named half, which returns a value that is half

that of the argument. (Assume the number variable references a float value.) Write

code for the function.

result = half(number)

3. A program contains the following function definition:

def cube(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.

4. Write a function named times\_ten that accepts a number as an argument. When the

function is called, it should return the value of its argument multiplied times 10.

5. Write a function named get\_first\_name that asks the user to enter his or her first

name, and returns it.

**Programming Exercises**

**1. Feet to Inches**

One foot equals 12 inches. Write a function named feet\_to\_inches that accepts a number

of feet as an argument, and returns the number of inches in that many feet. Use the

function in a program that prompts the user to enter a number of feet and then displays the

number of inches in that many feet.

**2. Math Quiz**

Write a program that gives simple math quizzes. The program should display two random

numbers that are to be added, such as:

247

+ 129

The program should allow the student to enter the answer. If the answer is correct, a message

of congratulations should be displayed. If the answer is incorrect, a message showing

the correct answer should be displayed.

**3. Maximum of Two Values**

Write a function named maximum that accepts two integer values as arguments and returns

the value that is the greater of the two. For example, if 7 and 12 are passed as arguments

to the function, the function should return 12. Use the function in a program that prompts

the user to enter two integer values. The program should display the value that is the

greater of the two.

**4. Falling Distance**

The following formula can be used to determine the distance an object falls due to gravity

in a specific time period, starting from rest:

*d* \_ 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 falling\_distance 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 calls the function in a loop that passes the

values 1 through 10 as arguments and displays the return value.

**5. Kinetic Energy**

In physics, an object that is in motion is said to have kinetic energy (KE). The following formula

can be used to determine a moving object’s kinetic energy:

*KE* \_ 1⁄2 *mv*2

The variables in the formula are as follows: *KE* is the kinetic energy in joules, *m* is the

object’s mass in kilograms, and *v* is the object’s velocity in meters per second.

Write a function named kinetic\_energy 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. Write a program that asks the user to enter values for

mass and velocity, and then calls the kinetic\_energy function to get the object’s kinetic

energy.

**6. Test Average and Grade**

Write a program that asks the user to enter five test scores. The program should display a letter

grade for each score and the average test score. Write the following functions in the program:

• calc\_average—This function should accept five test scores as arguments and return

the average of the scores.

• determine\_grade—This function should accept a test score as an argument and

return a letter grade for the score, based on the following grading scale:

**Score Letter Grade**

90–100 A

80–89 B

70–79 C

60–69 D

Below 60 F

**7. Odd/Even Counter**

In this chapter you saw an example of how to write an algorithm that determines whether

a number is even or odd. Write a program that generates 100 random numbers, and keeps

a count of how many of those random numbers are even and how many are odd.

**8. Prime Numbers**

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 Boolean function named is\_prime which takes an integer as an argument and

returns True if the argument is a prime number, or False otherwise. Use the function in a

program that prompts the user to enter a number and then displays a message indicating

whether the number is prime.

**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.

**9. Prime Number List**

This exercise assumes you have already written the is\_prime function in Programming

Exercise 8. Write another program that displays all of the prime numbers from 1 through 100.

The program should have a loop that calls the is\_prime function.

**10. Future Value**

Suppose you have a certain amount of money in a savings account that earns compound

monthly interest and you want to calculate the amount that you will have after a specific

number of months. The formula is

*F* \_ *P* \_ (*1* \_ *i*)*t*

The terms in the formula are as follows:

• *F* is the future value of the account after the specified time period.

• *P* is the present value of the account.

• *i* is the monthly interest rate.

• *t* is the number of months.

Write a program that prompts the user to enter the account’s present value, monthly interest

rate, and number of months that the money will be left in the account. The program should

pass these values to a function that returns the future value of the account after the specified

number of months. The program should display the account’s future value.

**11. Random Number Guessing Game**

Write a program that generates a random number in the range of 1 through 100 and asks the

user to guess what the number is. If the user’s guess is higher than the random number, the program

should display “Too high, try again.” If the user’s guess is lower than the random number,

the program should display “Too low, try again.” If the user guesses the number, the application

should congratulate the user and then generate a new random number so the game can

start over.

Optional Enhancement: Enhance the game so it keeps count of the number of guesses that the

user makes. When the user correctly guesses the random number, the program should display

the number of guesses.

**12. Rock, Paper, Scissors Game**

Write a program that lets the user play the game of Rock, Paper, Scissors against the computer.

The program should work as follows.

1. When the program begins, a random number in the range of 1 through 3 is generated.

If the number is 1, then the computer has chosen rock. If the number is 2, then the computer

has chosen paper. If the number is 3, then the computer has chosen scissors.

(Don’t display the computer’s choice yet.)

2. The user enters his or her choice of “rock”, “paper”, or “scissors” at the keyboard.

3. The computer’s choice is displayed.

4. A winner is selected according to the following rules:

• If one player chooses rock and the other player chooses scissors, then rock wins.

(The rock smashes the scissors.)

• If one player chooses scissors and the other player chooses paper, then scissors wins.

(Scissors cuts paper.)

• If one player chooses paper and the other player chooses rock, then paper wins.

(Paper wraps rock.)

• If both players make the same choice, the game must be played again to determine

the winner.