You can read the following textbook to find the answers:

- **Python for Everyone, 2nd Edition, Cay S. Horstmann, Rance D. Necaise, Wiley, 2016.**

1. Which process helps identify the functions that should make up a computer program?
   1. black boxing
   2. stepwise refinement
   3. parameter passing
   4. debugging

2. The term Black Box is used with functions because
   1. Only the implementation matters; the specification is not important.
   2. Only the specification matters; the implementation is not important.
   3. Only the arguments matter; the return value is not important.
   4. Only the return value matters; the arguments are not important.

3. One advantage of designing functions as black boxes is that
   1. many programmers can work on the same project without knowing the internal implementation details of functions.
   2. the result that is returned from black-box functions is always the same data type.
   3. the implementation of the function is open for everyone to see.
   4. there are fewer parameters.

4. A ____________________________ is a sequence of instructions with a name.

   1. variable
   2. argument
   3. parameter
   4. function

5. What is supplied to a function when it is called?

   1. arguments
   2. numbers
   3. return values
   4. variables

6. Consider the following function call |round(3.14159, 3)| what is the return value?

   1. |3.14159|
   2. |3.141|
   3. |3.14|
   4. |3.1|

7. Consider the following function call |ceil(3.14159)| what is the return value?

   1. |3.14159|
   2. |3.0|
   3. |4.0|
   4. |3.1416|

8. Which of the following is a correct call to Python's round function?

   1. |x = round("3.14159", 2)|
   2. |x = round("3.14159")|
   3. |x = round(3.14159)|
   4. |x = round(3, 1, 4, 1, 5, 9)|

9. Consider a function named |calc|. It accepts two integer arguments and returns their sum as an integer. Which of the following statements is a correct invocation of the calc function?

   1. total = calc()
   2. total = calc(2)
   3. total = calc("2", "3")
   4. total = calc(2, 3)

10. Which of the following statements correctly defines a function?

    1. |def functionName(parameterName1, parameterName2) :|
    2. |def functionName(parameterName1, parameterName2)|
    3. |functionName(parameterName1, parameterName2) :|
    4. |functionName(parameterName1, parameterName2)|

11. What Python statement exits a function and gives the result to the caller?

    1. |def|
    2. |return|

3. |send|
4. |result|

12. Given the code snippet below, what is returned by the function call:

|mystery(5,3)|?

```
def mystery(num1, num2) :
  result = num1 * num2
  return result
```

1. |8|
2. |15|
3. |2|
4. |0|

13. Given the code snippet below, what is returned by the function call:

|mystery(mystery(5, 3), mystery(5, 3))|?

```
def mystery(num1, num2) :
  result = num1 * num2
  return result
```

1. |225|
2. |15|
3. |30|

4. |0|

14. What is wrong with the following code snippet?

```
mystery(10, 2)
def mystery(num1, num2) :
   result = num1 ** num2
   return result
```

1. nothing, it will return |20|
2. nothing, it will return |100|
3. a variable must be used to store the result of the function call
4. the function must be defined before the statement that calls it

15. Consider the following function:

```
def w(x, y) :
   z = x + y
   return z
```

What is the function's name?

1. w
2. x
3. y
4. z

16. The following function is supposed to compute the area of a triangle and return the area as the function's result.

```
def triangleArea(base, height) :
  area = base * height / 2
  _____________________
```

What line of code must be placed in the blank to achieve this goal?

1. |print area|
2. |print(area)|
3. |return area|
4. |return triangleArea|

17. Assume that you are writing a function that computes the volume of a box for shipping electrical components. The components vary in shape -- some are long and skinny, while others are cube-like. Different boxes are used for components with different shapes. Which of the following function headers is the best?

1. |def boxVolume() :|
2. |def boxVolume(sideLength) :|
3. |def boxVolume(a, b, c) :|
4. |def boxVolume(length, width, height) :|

18. Consider the following function.

```
def factorial(n) :
  result = 1
  for i in range(1, n + 1) :
    result = result * i
  return result
```

What is the parameter variable for this function?

1. |factorial|
2. |i|
3. |n|
4. |result|

19. Consider the following function:

```
def squareArea(sideLength) :
  return sideLength ** 2
```

What is the value of squareArea(3)?

1. 2
2. 3
3. 6
4. 9

20. Consider the following function:

```
def squareArea(sideLength) :
   return sideLength ** 2
```

What is the value of squareArea(squareArea(2))?

1. 2
2. 4
3. 8
4. 16

21. Consider the following function:

```
def mystery(a, b) :
   result = (a - b) * (a + b)
   return result
```

What is the result of calling |mystery(3, 2)|?

1. 2
2. 3
3. 4
4. 5

22. Consider the following function:

```
## Compute the volume of a cuboid.
```

```
#  @param width the width of the cuboid
#  @return the volume of the cuboid
def volume(width, height, length) :
   return width * height * length
```

Based on the recommendations in the textbook, what change should be made to improve the comments for this function?

1. The |@param| line for width should be removed
2. Additional |@param| lines should be added for |height| and |length|
3. The first line should be expanded to describe how the function performs its calculation
4. The |@return| line should be removed

23. You are writing a function that converts from Liters to Gallons.

Which function header is the best?

1. |def litersToGallons() :|
2. |def litersToGallons(liters) :|
3. |def litersToGallons(gallons) :|
4. |def litersToGallons(liters, gallons) :|

24. Given the following code snippet, what is considered a parameter variable(s)?

```
def mystery(num1, num2) :
   result = num1 ** num2
   return result
```

```
mystery(10, 2)
```

1. |10, 2|
2. |num1, num2|
3. |result|
4. |mystery|

25. Given the following code snippet, what is considered an argument(s)?

```
def mystery(num1, num2) :
   result = num1 ** num2
   return result
mystery(10, 2)
```

1. |10, 2|
2. |num1, num2|
3. |result|
4. |mystery|

26. Parameter variables should not be changed within the body of a function because

1. This will generate a compiler error
2. This will generate a run-time error
3. It is confusing because it mixes the concept of a parameter with that of a variable
4. It is confusing because parameter variables cannot store values

27. Consider the following program:

```
def squareArea(sideLength) :
  return sideLength ** 2

a = squareArea(4)
```

What are the arguments (actual parameters) in this program?

1. 2
2. 4
3. sideLength
4. squareArea

28. Consider the following program:

```
def main() :
  a = 5
  print(doubleIt(a))

def doubleIt(x) :
  return x * 2

main()
```

What output is generated when this program is run?

1. 2
2. 4
3. 5
4. 10

29. Consider the following program:

```
def main() :
  a = 10
  print(doTwice(a))

def doTwice(x) :
  x = x * 2
  x = x * 2
  return x

main()
```

What output is generated when this program is run?

1. 2
2. 10
3. 20
4. 40

30. Consider the following program:

```
def main() :
  a = 2
  doubleIt(a)
  print(a)

def doubleIt(x) :
  x = x * 2

main()
```

What output is generated when this program is run?

1. 2
2. 4
3. 8
4. Python reports an error because doubleIt does not contain a
   return statement

31. Which statement causes the following function to exit immediately?

```
def mystery(num1, num2) :
  result = num1 ** num2
  return result
mystery(10, 2)
```

1. |mystery(10,2)|

2. |result = num1 ** num2|
3. |return result|
4. None of the statements cause the function to exit immediately.

32. Which statement should be added or modified to remove the possibility of a run-time error in this code snippet?

```
1: def floorDivision(value1, value2) :
2:    return value1 // value2
```

1. in line 2: change |//| to |/|
2. in line 2: change |//| to |%|
3. add this statement after line 1: |if value2 == 0 : return 0|
4. add this statement after line 1: |if value2 == 0 : return|

33. What happens in this code snippet if |sideLength = -10|?

```
def cubeSurfaceArea(sideLength) :
   if sideLength >= 0 :
      return 6 * (sideLength * sideLength)
# There are six sides to a cube; surface area of each side is sideLength squared
```

1. the function returns |600|
2. the function returns |-600|
3. an error occurs and aborts the program

4. a special value of |None| will be returned from the function

34. How many return statements can be included in a function?

1. Exactly one
2. One or two
3. One, two or more
4. Zero or more

35. What is the purpose of this code snippet?

```
def mystery(n) :
  if n % 2 == 0  :
    return True
  else :
    return False
```

1. to determine if |n| is even or odd
2. to find the remainder of |n| divided by 2
3. to find the value of |n| divided by 2
4. to determine if |n| is positive or negative

36. When should a computation be turned into a function?

1. when it may not be used
2. when it is only used once
3. when it may be used more than once

4. only if it contains complex mathematically equations

37. The following program is supposed to display a message indicating if the integer entered by the user is even or odd. What is wrong with the program?

```
num = int(input("Enter an integer: "))
print("The integer is", evenOdd(num))

def evenOdd(n) :
  if n % 2 == 0 :
    return "even"
  return "odd"
```

1. The function definition must appear before the function is called.
2. The input and print statements must reside in a function named |main|.
3. The variable |num| and the parameter variable |n| must have the same name.
4. An |else| clause must be added to the if statement.

38. The following function is supposed to return -1 when |x| is negative, +1 when |x| is positive, or 0 if |x| is zero. What, if anything, is wrong with the function?

```
def plusMinusZero(x) :
  if x == 0 :
    return 0
  elif x <= 0 :
    return -1
  else x >= 0 :
    return 1
```

1. A return statement must be added at the end of the function
2. Both occurrences of |elif| must be replaced with |if|
3. The |<=| and |>=| must be replaced with |<| and |>|
4. Nothing is wrong with the function

39. What is wrong with the following function for computing the amount of tax due on a purchase?

```
def taxDue(amount, taxRate) :
  amount = amount * taxRate

def main() :
  . . .
  total = taxDue(subtotal, TAX_RATE)
  . . .
```

1. The amount of tax due is not computed correctly
2. The function must print a value
3. The function must return a value

4. The function must take an additional parameter

40. The purpose of a function that does not return a value is

1. to package a repeated task as a function even though the task does not yield a value
2. to insert a temporary implementation of a function that can be refined later
3. to provide a function that can only be included in an assignment statement
4. only used when the function needs to produce output

41. Which function call correctly invokes the partial drawShape function listed below and prints a star triangle?

```
def drawShape(type) :
  length = len(type)
  if length == 0 :
    return
  if type == "triangle" :
    print("  *")
    print(" ***")
    print("*****")
drawShape("triangle")
```

1. |drawShape(triangle)|

2. |drawShape("triangle")|
3. |drawShape|
4. |value = drawShape(triangle)|

42. Consider the following functions:

```
def printIt(x) :
  print(x)

def incrementIt(x) :
  return x + 1

def decrementIt(x) :
  return x - 1

def doubleIt(x) :
  return x * 2
```

Which of the following function calls is *not* a reasonable thing to do?

1. print(printIt(5))
2. print(incrementIt(5))
3. print(decrementIt(5))
4. print(doubleIt(5))

43. The following function is supposed to compute and display the value of n-factorial for integers greater than 0.

```
def factorial(n) :
   result = 1
   for i in range(1, n + 1) :
     result = result * i
```

What is wrong with this function?

1. The indenting is wrong. All of the lines should be indented by the same amount.
2. The calculation is wrong. The |result| variable will have something other than n-factorial stored in it.
3. The function is missing a line. A |print| statement must be added at the end of it.
4. The function is missing a line. A |return| statement must be added at the end of it.

44. The function below randomly generates a number between 1 and 6 to represent a single die. Which implementation listed below allows for other types of die?

```
def die() :
   return randint(1, 6)
```

1.

```
def die(low, high) :
    return
```

2.

```
def die(low, high) :
    return randint(low, high)
```

3.

```
def die(high) :
    return randint(0, high)
```

4.

```
def die(low, high) :
    return high % low
```

45. Given these two separate functions, which implemenation combines them into one reusable function?

4.

```
def sixSidedDie() :
  return randint(1, 6)
def fourSidedDie() :
  return randint(1, 4)
```

1.

```
def die(low, high) :
  return
```

2.

```
def die(low, high) :
  return high % low
```

3.

```
def die(high) :
  return randint(0, high)
```

4.

```
def die(low, high) :
   return randint(low, high)
```

46. What is stepwise refinement?

1. The process of unit testing
2. The design of pseudocode for black-box functions
3. The process of breaking complex problems down into smaller, manageable steps
4. The use of a temporary implementation of a function that can be improved later

47.

Why is hand-tracing or manually walking through the execution of a function helpful?

1. It enforces the "black-box" concept of function design
2. It makes unit testing unnecessary
3. It guarantees that the function will compile without errors
4. It is an effective way to understand a function's subtle aspects

48.

When hand-tracing functions, the values for the parameter variables:

1. Need not be traced because they are never returned
2. Are the same each time the function is invoked
3. May be undetermined or missing when the function executes
4. Are determined by the arguments supplied in the code that
   invokes the function

49.

A stub function is

1. A short function
2. A function that has been unit tested
3. A function that acts as a placeholder and returns a simple value
   so another function can be tested
4. A function that is broken down into smaller steps through
   stepwise refinement

50. Consider the following function:

```
def numberToGrade(x) :
  return "X"
```

This function will eventually be rewritten so that it returns the letter grade associated with |x| grade points. However, at the moment it is incomplete, and always returns the letter |X| as a placeholder. In its current form, this function is referred to as a:

1. def
2. param
3. refinement
4. stub

51.

The variable name |perfect| in the function |myFun| in the code snippet below is used as both a parameter variable and a variable in the body of the function. Which statement about this situation is true?

```
def myFun(perfect)
  perfect = 0
  return ((perfect - 1) * (perfect - 1))
```

1. This multiple use of the same variable |perfect| will not compile because the scopes overlap
2. While this is legal and will compile in Python, it is confusing
3. Because the scopes of these variables do not overlap, there is no problem
4. This situation rarely occurs and the compiler always issues a warning

52.

Consider the following code segment:

```
def main() :
  avg = 0
  total = 0
  for i in range(6) :
    iSquared = i * i
    total = total + iSquared
    avg = total / i
  print(total)
  print(avg)
```

Which of the following answers lists all of the local variables in this code segment?

1. |avg, total, i, iSquared|
2. |i, iSquared|
3. |avg, total|
4. |i|

53.

Given the following code snippet, which statement correctly allows the function to update the global variable |total|?

```
1. total = 0
2. def main() :
3.   avg = 0
```

```
4.   for i in range(6) :
5.      iSquared = i * i
6.      total = total + iSquared
7.   avg = total / i
8.   print(total)
9.   print(avg)
```

1. add the keyword |global| to line 1
2. line 1 already allows the |total| variable to be updated
3. move line 1 inside the function definition
4. add the statement |global total| after line 2

54.

What is the output from the following Python program?

```
def myFun(perfect) :
   perfect = 0
   return ((perfect - 1) * (perfect - 1))

def main() :
   for i in range(4) :
       print(myFun(i), end = " ")

main()
```

1. |1 1 1 1|
2. |-1 0 1 4|
3. |0 0 0 0|
4. |1 0 1 4|

55.

What is the output from the following Python program?

```
def main() :
  a = 10
  r = cubeVolume()
  print(r)

def cubeVolume() :
  return a ** 3

main()
```

1. |10|
2. |30|
3. |1000|
4. Nothing, there is an error.

56.

Which line of code in the Python program below is the recursive invocation of function |myFun?|

```
1: def main() :
2:    for i in range(4) :
3:       print(myFun(i), end = " ")
4: def myFun(perfect) :
5:    perfect = 0
6:    return ((perfect - 1) * (perfect - 1))
7: main()
```

1. Line 1
2. Line 3
3. Line 6
4. There is no recursive invocation in this code segment

57.

Which of the following is NOT a good practice when developing a computer program?

1. Put as many statements as possible into the main function
2. Document the purpose of each function parameter
3. Decompose a program into many small functions
4. Place code that is used multiple times into a separate function

58.

Which of the following statements about variables is true?

1. A variable is visible from the point at which it is defined
   until the end of the program.
2. You should use global variables whenever possible..
3. The same name can be used for two different variables in a
   single method.
4. The same variable name can be used in two different methods.

59. The |ceil| function in the Python standard library |math| module
takes a single value x and returns the smallest integer that is
greater than or equal to x. Which of the following is true about
|ceil(56.75)|?

1. The argument is 56.75, and the return value is 57
2. The argument is 56.75, and the return value is 56
3. The argument is 57, and the return value is 56.75
4. The argument is 56, and the return value is 56.75

60. Consider a function named |avg|, which accepts four numbers and
returns their average. Which of the following is a correct call to
the function |avg|?

1. |avg(2, 3.14, 3, 5, 6)|
2. |average = avg(2, 3.14, 4, 5)|

3. |avg()|
4. |average = avg("2", "3", "4", "5")|

61. Which of the following statements is true about functions in Python?

1. Functions can have only one argument and can return only one return value.
2. Functions can have multiple arguments and can return multiple return values.
3. Functions can have multiple arguments and can return one return value.
4. Functions can have one argument and can return multiple return values.

62.

A programmer notices that the following code snippet uses the same algorithm for computing interest earned, but with different variables, in the two places shown below and in several other places in the program. What could be done to improve the program?

```
RATE1 = 10
RATE2 = 5.5
interest = investment * RATE1 / 100
. . .
balance = balance + balance * RATE2 / 100
```

1. Declare the rates as variables, not constants.
2. Define a function that looks up interest rates.
3. Define a function that prompts the user for an amount and a rate of interest, then returns the interest earned.
4. Define a function that computes the interest earned from an amount and a rate of interest.

63. What is wrong with the following code?

```
def grade(score) :
  if score >= 90 :
    return "A"
  elif score >= 80 :
    return "B"
  elif score >= 70 :
    return "C"
  elif score >= 60 :
    return "D"
```

1. The name of the parameter variable is illegal
2. The type of the parameter variable is invalid
3. Another |return| statement needs to be added to the function
4. One of the existing |return| statements is not correct

64. Given the following code, what is the output?

```
def main() :
  i = 20
  b = mysteriousFunction2(i)
  print(b + i)

def mysteriousFunction1(i) :
  n = 0
  while n * n <= i :
    n += 1
  return n - 1

def mysteriousFunction2(a) :
  b = 0
  for n in range(a) :
    i = mysteriousFunction1(n)
    b = b + i
  return b

main()
```

1. 50
2. 60
3. 70
4. 80

65. For a program that reads city names repeatedly from the user and

calculates the distance from a company's headquarters, which of the following would be a good design based on stepwise refinement?

1. Write on function that calculates distance randomly
2. Write one function that reads city name
3. Write one function that reads city name and another function that calculates distance
4. Write one function that reads distance and finds city name

66. For a program that reads three letter grades and calculates an average of those grades, which of the following would be a good design based on stepwise refinement?

1. Write one function that reads three letter grades, converts each letter grade to a number, and calculates the average of the three numbers.
2. Write one function that reads three letter grades, and a second function to convert each letter to a number and calculate the average of the three numbers.
3. Write one function that reads a letter grade and returns the number equivalent, and one function that computes the average of three numbers.
4. Stepwise refinement cannot be applied to this problem.

67. What is the output of the following code snippet?

```
def main() :
   print(blackBox(4))
```

```
def blackBox(a) :
  if a <= 0 :
    val = 1
  else :
    val = a + blackBox(a - 2)
  return val

main()
```

1. 4
2. 2
3. 1
4. 7

68. What is the output of the following code snippet?

```
def main() :
  print("fun(2) =", fun(2))

def fun(a) :
  returnValue = 0
  if a > 5 :
    returnValue = a
  else :
    returnValue = fun(2 * a)
  return returnValue
```

```
main()
```

1. |fun(2) = 4|
2. |fun(2) = 8|
3. |fun(2) = 16|
4. |fun(2) = 32|

69.

Based on the code snippet, which of the following statements is correct?

```
def main() :
  reoccur(1)

def reoccur(a) :
  print(a)
  reoccur(a + 1)

main()
```

1. The code snippet gives a compilation error as the reoccur function cannot call itself.
2. The code snippet executes and infinitely recurses, displaying 1, 2, 3, 4, and so on.
3. The code snippet executes and displays 1.
4. The code snippet executes and does not produce any output.

70.

What is the output if the function call is |testMyVal(6)| in the following code snippet?

```
def testMyVal(a) :
   if a > 0 :
      testMyVal(a - 2)
   print(a, end = " ")
```

1. 0 2 4 6
2. 0 0 0 0
3. The code snippet executes and infinitely recurses, displaying 2, 4, 5, and so on.
4. 6 4 2 0

71. Which of the following code snippets returns the factorial of a given number? (Hint: Factorial of 5 = 5! = 1 * 2 * 3 * 4 * 5 = 120)

1.

```
def factorial(num) :
   return num * factorial(num - 1)
```

2.

```
def factorial(num) :
  if(num == 1) :
    return 1
  else :
    return num * factorial(num)
```

3.

```
def factorial(num) :
  if(num == 1) :
    return 1
  else :
    print(num * factorial(num - 1))
```

4.

```
def factorial(num) :
  if(num == 1) :
    return 1
  else :
    return num * factorial(num - 1)
```

72. Consider this recursive function:

```
def mystery(x) :
  if x <= 0 :
    return 0
  else :
    return x + mystery(x - 1)
```

What is mystery(5)?

1. 15
2. 10
3. 0
4. 4

73. In the code snippet below, which variables are considered global variables:

```
a = 0
b = 5
def main() :
  global a, b
  fun1()
  fun2()

def fun1() :
  i = 0
  b = 0
```

```
def fun2() :
  a = b + 1

main()
```

1. |a, b, i|
2. |fun1, fun2|
3. |a, b|
4. |i|

74. Consider the following program:

```
def main() :
  print(factorial(n))        # Line 1

def factorial(n) :           # Line 2
  result = 1
  for i in range(1, n + 1) :
    result = result * i      # Line 3
  return i

main()
```

Which of the following lines is a recursive function call?

1. Line 1
2. Line 2
3. Line 3

4. There are no recursive function calls in the program

75. Consider the following program:

```
def main() :
   print(factorial(n))          # Line 1

def factorial(n) :              # Line 2
   if n <= 1 :
      return 1
   return n * factorial(n - 1)    # Line 3

main()
```

What line is the recursive call on?

1. Line 1
2. Line 2
3. Line 3
4. There are no recursive function calls in the program

76. What output is generated when the following program runs?

```
def main() :
   x = mystery(5)
   print(x)
```

```
def mystery(n) :
  if n == 1 :
    return 1
  return 2 * mystery(n - 1)

main()
```

1. 4
2. 8
3. 16
4. 32

77. What output is generated when the following program runs?

```
def main() :
  x = mystery(9, 12)
  print(x)

def mystery(a, b) :
  if b == 0 :
    return a
  else :
    return mystery(b, a % b)

main()
```

1. 3
2. 6
3. 9
4. 12

78. The tool that allows you to follow the execution of a program and helps you locate errors is known as a(n):

1. Compiler
2. Debugger
3. Interpreter
4. Virtual machine

79. The location where the debugger stops executing your program so that you can inspect the values of variables is called a(n):

1. breakpoint
2. function call
3. inspection point
4. step point

80. Which debugging command allows you to quickly run an entire function instead of examining its body a line at a time?

1. Step into
2. Step over
3. Step point
4. Step quickly

81. What term is used to refer to a collection of functions and classes organized into one or more user-defined modules?

1. function argument
2. hardware toolbox
3. software toolkit
4. standard library

82. Which function should *not* be included in a software toolkit for managing student enrollment in courses?

1. |computeNegativeImage|
2. |enrollStudentInCourse|
3. |getGradePointAverage|
4. |setFinalExamMark|

83. Consider the following code segment:

```
def f1():
  print("A", end="")

def f2():
  f1()
  print("B", end="")
```

What output is generated when it runs?

1. |A|
2. |AB|
3. |B|
4. The code segment does not display any output.

84. Consider the following code segment:

```
def f1():
  print("a", end="")
  return "b"

def f2():
  print("c", end="")
  d = f1()
  print(d, end="")
  print("e", end="")

def f3():
  print("f", end="")
  f2()
  print("g", end="")

f3()
```

What output is generated when it runs?

1. |fg|

2. |fceg|
3. |fcabeg|
4. |fcadeg|

85. What term is used to describe the portion of a program in which a variable can be accessed?
    1. locale
    2. region
    3. scope
    4. volume