You can read the following textbook to find the answers:

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

1. To store a value for later use in Python, the programmer needs to create a:
    1. number
    2. character
    3. variable
    4. boolean
2. A variable is:
    1. A storage location with a name
    2. An assignment statement
    3. An expression
    4. A point in a program where a decision is made
3. How is a value stored in a variable?
    1. an assignment statement
    2. an expression
    3. a print statement
    4. an equality statement
4. What is the value of the variable `num` after the following code snippet?

```
num = 5
num2 = 6
num = num2 + 3
```

    1. 5
    2. 9
    3. 8
    4. 11
5. What is wrong with this assignment statement?

```
num + 10 = num2
```

    1. The left hand side of an assignment statement cannot include an operator.
    2. Nothing, this statement compiles and executes.
    3. The value of `10` must be defined before this statement can be executed.
    4. The `num` variable must be defined before this statement can be executed
6. What is the right way to assign the value of `num + 10` to `num2`?
    1. `num2 = num + 10`
    2. `num = num2 + 10`
    3. `num2 + 10 = num`
    4. `num + 10 = num2`
7. What is wrong with the following code snippet?

```
num = 78A
```

1. The `num` variable is never assigned a value
2. `78A` is not a valid value in a Python program
3. The name `num` is not a valid variable name
4. The `num` variable is never initialized

8. What is wrong with the following code snippet?

```
2ndNum = 78
```

1. The `2ndNum` variable is never assigned a value
2. The `2ndNum` variable is assigned a non-numeric value
3. The `2ndNum` variable is not a valid variable name
4. The `2ndNum` variable is never initialized

9. What is a variable called that should remain unchanged throughout your program?
    1. a constant variable
    2. a data variable
    3. a string variable
    4. a boolean variable
10. Which of the following variables should be coded as a constant in Python?
    1. character: 'a'
    2. string: "hello"
    3. number: 1234
    4. pi: 3.14159
11. Which of the following variable names follows the Python naming convention for constants?
    1. `maxSize`
    2. `MAX_SIZE`
    3. `MAX SIZE`
    4. `max_size`
12. Why is it important to follow Python naming standards for variables representing constants?
    1. it is good programming style
    2. it is required by the Python programming language
    3. it is required by graphic programs
    4. it is required for all non-zero numbers
13. Which of the following symbols can be used to begin a string literal in Python?
    1. `*`
    2. `#`
    3. `"`
    4. `>`
14. Which of the following is an appropriate constant name to represent the number of pencils in a pack?
    1. `NUM_PENCILS_PER_PACK = 12`
    2. `numPencilsPerPack = 12`
    3. `NUMpencilsPERpack = 12`

    4. `numpencilsperpack = 12`
15. Which line of code creates a variable named x and initializes it to the integer 5?
    1. `x = 5.0`
    2. `x = 5`
    3. `x = '5'`
    4. `x = "5"`
16. Which of the following names is **not** a legal variable name?
    1. `bottle-volume`
    2. `cans_per_pack`
    3. `four`
    4. `x2`
17. Which of the following names is the best for a constant variable holding the price of a can of soda?
    1. `soda_price`
    2. `soda-price`
    3. `SodaPrice`
    4. `SODA_PRICE`
18. What symbol is used to begin a comment in a Python program?
    1. `!`
    2. `@`
    3. `#`
    4. `$`
19. Which of the following statements correctly multiplies `num1` times `num2`?
    1. `num1 * num2`
    2. `num1 x num2`
    3. `num1 · num2`
    4. `num1 ** num2`
20. Which of the following statements correctly calculates the average of three numbers: `num1, num2, and num3`?
    1. `num1 + num2 + num3 / 3`
    2. `num1 + num2 + num3 % 3`
    3. `( num1 + num2 + num3 ) / 3`
    4. `( num1 + num2 + num3 / 3 )`
21. Which of the following suggestions is the best way to make code easier for other programmers to understand?
    1. Use more statements in the source code.
    2. Give each variable a name that explains its purpose.
    3. Avoid using complex calculations in the source code
    4. Use single-letter variable names in the source code
22. What is the value of `result` after the following code snippet?

```
num1 = 10
num2 = 20
num3 = 2
result = num1 / num2 / num3
print(result)
```

1. 1
2. 0
3. The code has an error
4. 0.25

23. What will be the values of the variables `num1` and `num2` after the given set of assignments?

```
num1 = 20
num2 = 10
num1 = num1 + num2 / 2
num2 = num1
```

1. `num1 = 20.0, num2 = 10.0`
2. `num1 = 15.0, num2 = 10.0`
3. `num1 = 25.0, num2 = 25.0`
4. `num1 = 15.0, num2 = 15.0`

24. What is the value of `result` after the following code snippet?

```
num1 = 10
num2 = 20
num3 = 2
result = num1 // num2 // num3
print(result)
```

1. 1
2. 0
3. The code has an error
4. 0.25

25. What is the value of `result` after the following code snippet?

```
num1 = 20
num2 = 10
num3 = 2
result = num1 // num2 // num3
print(result)
```

1. 1
2. 0
3. The code has an error
4. 0.25

26. What is the value of `result` after the following code snippet?

```
num1 = 20
num2 = 10
num3 = 2
result = num1 // num2 / num3
print(result)
```

1. 1.0
2. 0.0

3. The code has an error
4. 0.25

27. Assume that you have an integer variable, `pennies`, that currently contains an integer number of pennies. Which statement determines the number of dollars and cents for that number of pennies?

```
1. dollars = pennies // 100
2. cents = pennies % 100
3. dollars = pennies / 100
4. cents = pennies % 100
5. dollars = pennies // 100
6. cents = pennies / 100
7. dollars = pennies % 100
8. cents = pennies / 100
```

28. Which code snippet is the correct Python equivalent to the following Algebraic expression ?

$c = \sqrt{(a^2 + b^2)}$

1. `sqrt(a ^ 2 + b ^ 2)`
2. `sqrt(a ** 2 + b ** 2)`
3. `sqrt(a * 2 + b * 2)`
4. `squareroot(a ** 2 + b ** 2)`

29. What symbol is used to find remainder of a floor division?
    1. //
    2. `/`
    3. %
    4. #
30. A(n) _____ is a collection of programming instructions that carry out a particular task.
    1. argument
    2. parameter
    3. function
    4. literal
31. What is the value of `4 ** 3`?
    1. `12`
    2. `64`
    3. `1`
    4. Nothing, there is an error in the statement
32. What is returned by the function: `round(x)` if `x = 5.64`?
    1. Nothing, there is an error in the statement
    2. `5`
    3. `5.6`
    4. `6`
33. What is returned by the function: `abs(x)` if `x = 5.64`?
    1. Nothing, there is an error in the statement
    2. `5`
    3. `5.64`

4. `6`

34. What is returned by the function: `max(1, 4, 15, 2, 3, 24)`?
    1. `1`
    2. `24`
    3. `15`
    4. `2`
35. What is returned by the function: `round(3.14159, 2)`?
    1. `3`
    2. `3.14159`
    3. `3.2`
    4. `3.14`
36. What is wrong with the following code snippet?

```
result = num1 // num2 / num3
num1 = 20
num2 = 10
num3 = 2
print(result)
```

    1. A variable is used before it is assigned a value.
    2. Nothing, the code compiles and runs.
    3. The `//` symbol cannot be used in a Python program.
    4. One or more of the variable names is not valid.
37. A(n) _____ is a collection of code that has been written by someone else that is ready for you to use in your program.
    1. variable
    2. argument
    3. function
    4. library
38. What must be done first before you can use a function from the standard library?
    1. the function must be defined
    2. the function must be imported
    3. the function must be included in a module
    4. the function must be enclosed in parenthesis
39. What is returned by the function: `sqrt(64)`?
    1. `8.0`
    2. `32.0`
    3. `4.0`
    4. `64.0`
40. What is wrong with the following code snippet?

```
((num1 + num2) * num3 / 2 * (1 - num4)
```

    1. nothing, the code compiles and runs
    2. there is an extra parenthesis
    3. parenthesis are not required
    4. illegal expression

41. Consider the following code segment:

```
x = 5
y = 7
z = x - y * 2
```

After this code segment executes, the value of z is:

1. -9
2. -4
3. 5
4. 7

42. The Python code that represents the formula $c = (a / b)^3$ is:
    1. `c = a / b ** 3`
    2. `c = (a / b) ** 3`
    3. `c = 3 ^ (a / b)`
    4. `c = (a / b) ^ 3`
43. Consider the following code segment:

```
x = 5
y = 3
z = 2
result = x // y + x % z
```

After this code segment, the value of `result` is:

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

44. Which statement computes the square root of 5 and stores it in the variable, r? Assume that the `math` module has already been imported.
    1. `r = math.squareRoot(5)`
    2. `r = math.sqrt(5)`
    3. `r = math.squareRoot[5]`
    4. `r = math.sqrt[5]`
45. Which of the following statements computes the minimum of the variables `a`, `b`, `c` and `d`, and stores it in `x`?
    1. `x = minimum(a, b, c, d)`
    2. `x = min(a, b, c, min(d))`
    3. `x = min(min(a, b), min(c, d))`
    4. `min(a, b, c, d) = x`
46. A sequence of characters is referred to as a:
    1. string
    2. module
    3. variable

4. expression

47. What is the value of `length` after this statement: `length = len("Good Morning")?`
    1. `10`
    2. `11`
    3. `12`
    4. `13`
48. What is it called when you join two strings together in Python?
    1. concatenation
    2. addition
    3. repetition
    4. conversion
49. Which statement correctly creates a new variable by combining the two string variables: `firstName` and `lastName`?
    1. `name = "firstName" + "lastName"`
    2. `name = firstName + lastName`
    3. `name = first name + last name`
    4. `name = firstName & lastName`
50. What is printed from the following code snippet:

```
message = "ho.."
print(message * 3)
```

    1. `ho..ho..ho`
    2. `ho..`
    3. `ho..ho..ho..`
    4. nothing is printed, this code snippet causes an error
51. What is printed by the following code snippet:

```
street = " Main Street"
address = 123 + street
print(address)
```

    1. 123Main Street
    2. 123 Main Street
    3. 123 "Main Street"
    4. nothing is printed, this code snippet causes an error
52. The following code snippet has an error, how can this be corrected so it prints: 123 Main Street?

```
1. street = " Main Street"
2. address = 123 + street
3. print(address)
```

    1. change the value '123' in line 2 to a string using the `str` function
    2. reverse lines 1 and 2
    3. change line 1 to read: `street = 123 + "Main Street"`
    4. change line 2 to read: `address = 123 + "Main Street"`

53. What functions can be used to convert a string into a number?
    1. `stri` and `len`
    2. `int` and `float`
    3. `sqrt`, `abs` and `round`
    4. `integer` and `float`
54. What is printed by the following code snippet?

```
num = int("45") * float("1.5")
print(num)
```

    1. nothing, this causes an error
    2. `46.5`
    3. `45 * 1.5`
    4. `67.5`
55. What is the index value of the letter `'h'` in the string below ?

```
message = "hello"
```

    1. `1`
    2. `0`
    3. `3`
    4. `4`
56. Given the code snippet below, what code is needed to print the person's initials?

```
firstName = "Pamela"
middleName = "Rose"
lastName = "Smith"
```

    1. `print(firstName[1], middleName[1], lastName[1])`
    2. `print(firstName[0], middleName[0], lastName[0])`
    3. `print(firstName + middleName + lastName)`
    4. `print(firstName, middleName, lastName)`
57. What output is generated by the following code snippet?

```
firstName = "Pamela"
middleName = "Rose"
lastName = "Smith"
print(firstName[0], middleName[0], lastName[5])
```

    1. nothing, this causes an index of bounds error
    2. `PRh`
    3. `P R h`
    4. `PRS`
58. Which statement finds the last letter of the string variable `name?`
    1. `last = name[len(name)]`
    2. `last = len(name) - 1`
    3. `last = len(name)`

4. `last = name[len(name) - 1]`

59. A _____ is a collection of programming instructions that can be appllied to an object.
    1. function
    2. method
    3. class
    4. object
60. What is printed by the following code snippet?

```
name = "Robert"
formalName = name.upper()
print(formalName)
```

    1. Robert
    2. robert
    3. ROBERT
    4. formalName
61. What is printed by the following code snippet?

```
name = "Robert"
formalName = name.lower()
print(formalName)
```

    1. Robert
    2. robert
    3. ROBERT
    4. formalName
62. What is printed by the following code snippet?

```
name = "today is thursday"
name.replace("t", "T")
name.replace("i", "I")
print(name)
```

    1. `today is thursday`
    2. `Today is Thursday`
    3. `Today Is Thursday`
    4. `Today Is thursday`
63. What is printed by the following code snippet?

```
name = "today is thursday"
newName = name.replace("t", "T")
print(newName)
```

    1. `today is thursday`
    2. `Today is Thursday`
    3. `Today Is Thursday`
    4. `Today Is thursday`

64. What is the value of x after the following code segment?

```
x = len("Hello World!")
```

    1. 10
    2. 11
    3. 12
    4. 13

65. Assume that `s` is an arbitrary string containing at least 2 characters. What is displayed by the following code segment?

```
print(s[0], s[len(s) - 1])
```

    1. The first character of `s`, followed immediately by the second last character of `s`.
    2. The first character of `s`, followed immediately by the last character of `s`.
    3. The first character of `s`, followed by a space, followed by the second last character of `s`.
    4. The first character of `s`, followed by a space, followed by the last character of `s`.

66. What is the value of `words` after the following code segment?

```
words = "Hello" + "World" * 3
```

    1. `"HelloWorldWorldWorld"`
    2. `"Hello World World World"`
    3. `"HelloWorldHelloWorldHelloWorld"`
    4. `"Hello World Hello World Hello World"`

67. Which of the following statements causes Python to report an error?
    1. `x = 17 + 18.4`
    2. `x = 17 + "18.4"`
    3. `x = 17 + int(18.4)`
    4. `x = 17 + float("18.4")`

68. What letter is displayed by the following code segment?

```
title = "Python for Everyone"
print(title[3])
```

    1. `e`
    2. `h`
    3. `o`
    4. `t`

69. Consider the following code segment:

```
product = "Cookies"
```

```
product = product.lower()
```

After this code segment executes, the value of the `product` variable is:

1. `"cookies"`
2. `"cOOKIES"`
3. `"Cookies"`
4. `"COOKIES"`

70. Consider the following code segment:

```
title = "Python for Everyone"
newTitle = title.replace("e", "*")
```

After this code runs, the value stored in `newTitle` is:

1. `"Python for *veryone"`
2. `"Python for Ev*ryone"`
3. `"Python for Ev*ryon*"`
4. `"Python for *v*ryon*"`

71. What is displayed by the following code segment?

```
print("\"Hello World!\"")
```

1. `Hello World!`
2. `"Hello World!"`
3. `\"Hello World!\"`
4. The program reports an error

72. What function is used to read a value from the keyboard?
    1. `input`
    2. `print`
    3. `keyboard`
    4. `next`
73. The message used to tell the user what input is expected is known as a(n):
    1. input
    2. keyword
    3. comment
    4. prompt
74. What is the data type of the value returned by the input function?
    1. integer
    2. string
    3. float
    4. character
75. Which statement correctly saves the price in the variable `cost`?

```
userInput = input("Please enter the price:")
```

1. `cost = float(userInput)`
2. `cost = userInput`
3. `cost = int(userInput)`
4. `cost = float[userInput]`

76. Which statement correctly saves the number of items in the variable `quantity`?

```
userInput = input("Please enter the quantity:")
```

1. `quantity = float(userInput)`
2. `quantity = userInput`
3. `quantity = int(userInput)`
4. `quantity = int[userInput]`

77. What is printed by the following code snippet?

```
cost = 25.45378
print("%.2f" % cost)
```

1. `25.45378`
2. `%25.45`
3. `25.45`
4. nothing, there is an error

78. Which output format string correctly allows for 5 positions before and two digits after the decimal point?
    1. `"%8.2f"`
    2. `"%5.2f"`
    3. `"%7.2f"`
    4. `"%5d.2f"`
79. Which output format correctly prints an item description left justified with up to 10 letters?
    1. `"%10"`
    2. `"%10s"`
    3. `"%-10s"`
    4. `"-%10s"`
80. What is the output for the following code snippet:

```
area = 25
print("The area is %05d" % area)
```

1. `The area is 25`
2. nothing, there is an error in the code snippet
3. `The area is 00025`
4. `The area is 25`

81. Consider the following code segment:

```
a = input("Enter the value of a: ")
b = input("Enter the value of b: ")
```

```
print(a + b)
```

When this code segment is run the user enters 1 at the first prompt and 5 at the second prompt. The output displayed is:

1. `1`
2. `6`
3. `15`
4. `1 + 5`

82. The line of code which reads a value from the user and stores it in a variable named `x` as a floating-point value is:
    1. `x = float()`
    2. `x = input("Enter the value of x: ")`
    3. `x = float(input("Enter the value of x: "))`
    4. `x = input(float())`
83. The line of code that displays the floating point number stored in the variable `x` using 3 decimal places is:
    1. `print("%.3f", x)`
    2. `print("%.3f" % x)`
    3. `print("%3.f", x)`
    4. `print("%3.f" % x)`
84. What output is generated by the following code segment?

```
a = 10.0
b = 0.50
print("The total is %.2f and the tax is %.2f." % (a, b))
```

    1. `The total is .00 and the tax is .50`
    2. `The total is 10.0 and the tax is 0.5`
    3. `The total is 10.0 and the tax is 0.50`
    4. `The total is 10.00 and the tax is 0.50`
85. Consider the following code segment:

```
x = 12
print("%d%%" % x)
```

The output generated by this code segment is:

1. `12`
2. `%12`
3. `12%`
4. `12%%`

86. A graphics application shows information inside a _____
    1. panel
    2. window

3. form
4. page

87. Which statement draws a square on the canvas?
    1. `canvas.drawRect(0, 50, 0, 50)`
    2. `canvas.drawRect(50, 50, 0, 0)`
    3. `canvas.drawRect(0, 0, 50, 100)`
    4. `canvas.drawRect(0, 0, 50, 50)`
88. Which statement sets the fill color when drawing shapes on the canvas?
    1. `canvas.setOutline("black")`
    2. `canvas.setFill("black")`
    3. `canvas.fill("black")`
    4. `canvas.fillRect("black")`
89. Which statement writes the word `Hello` on the canvas?
    1. `canvas.setString(10, 10, "Hello")`
    2. `canvas.setText(10, 10, "Hello")`
    3. `canvas.drawText(10, 10, "Hello")`
    4. `canvas.drawString(10, 10, "Hello")`
90. Which of the given print statements generates the following output?

```
ABCDE"\
```

    1. `print("ABCDE\"\\")`
    2. `print("ABCDE"\")`
    3. `print("ABCDE"\)`
    4. `print("ABCDE\"\")`
91. Consider the following code segment:

```
from graphics import GraphicsWindow

win = GraphicsWindow(400, 200)
canvas = win.canvas()
```

    The line of code that should be added to the end of the code segment above to draw a diagonal line connecting the upper left corner to the lower right corner is:

    1. `canvas.drawLine(0, 0, 0, 0)`
    2. `canvas.drawLine(0, 0, 200, 400)`
    3. `canvas.drawLine(200, 400, 400, 200)`
    4. `canvas.drawLine(400, 200, 0, 0)`
92. The statement that sets the fill color to red is:
    1. `canvas.setFill(0, 128, 0)`
    2. `canvas.setFill(64, 0, 128)`
    3. `canvas.setFill(64, 255, 64)`
    4. `canvas.setFill(128, 0, 0)`
93. Which of the following statements draws a circle?
    1. `canvas.drawOval(100, 200, 100, 200)`

2. `canvas.drawOval(200, 100, 200, 200)`
3. `canvas.drawOval(200, 200, 100, 200)`
4. `canvas.drawOval(200, 200, 200, 100)`

94. Which statement imports the entire contents of the `sympy` module?
    1. `from sympy import *`
    2. `import contents`
    3. `import * from sympy`
    4. `sympy import`
95. Which statement creates an expression in SymPy form?
    1. `f = x ** 2`
    2. `f = sympify("x ** 2")`
    3. `f = sympy(x ** 2)`
    4. `sympy("f = x ** 2")`
96. Which SymPy function is used to display a graph of a mathematical function?
    1. `diff`
    2. `draw`
    3. `plot`
    4. `subs`
97. Which of the following items is an example of a floating-point literal?
    1. `100000`
    2. `100,000`
    3. `100000.0`
    4. `100,000.0`
98. Which of the following statements about variable names is **not** correct?
    1. Variable names are case sensitive.
    2. Variable names can begin with a letter, an underscore or a number.
    3. Variable names cannot be reserved words such as `if` and `class`.
    4. Variable names cannot contain symbols such as `?` and `%`.
99. What convention is normally used when naming constants in a Python program?
    1. Constant names are normally written in all capital letters.
    2. Constant names normally begin with a `#` character.
    3. Constant names normally begin and end with an underscore.
    4. Constant names normally begin with a capital letter followed only by numbers.
100. A numeric constant that appears in your code without explanation is known as a:
    1. floating-point number
    2. magic number
    3. string
    4. variable
101. Which function call will cause Python to report an error?
    1. `abs(1, 2)`
    2. `max(1, 2)`
    3. `min(1, 2)`
    4. `round(1, 2)`
102. Which statement causes `A` and `B` to be printed on different lines?
    1. `print("AB")`
    2. `print("A,B")`

3. `print("A\nB")`
4. `print("A", "B")`