**Starting Out with Python 4e (Gaddis)**

**Chapter 5 Functions**

**TRUE/FALSE**

1. Python function names follow the same rules as those for naming variables.

ANS: T

2. The function header marks the beginning of the function definition.

ANS: T

3. A function definition specifies what a function does and causes the function to execute.

ANS: F

4. A hierarchy chart shows all the steps that are taken inside a function.

ANS: F

5. A local variable can be accessed from anywhere in the program.

ANS: F

6. Different functions can have local variables with the same names.

ANS: T

7. Python allows you to pass multiple arguments to a function.

ANS: T

8. To assign a value to a global variable in a function, the global variable must be first declared in the function.

ANS: T

9. The value assigned to a global constant can be changed in the mainline logic.

ANS: F

10. One reason not to use global variables is that it makes a program hard to debug.

ANS: T

11. A value-returning function is like a simple function except that when it finishes it returns a value back to the part of the program that called it.

ANS: T

12. Unlike other languages, in Python the number of values a function can return is limited to one.

ANS: F

13. In Python you can have a list of variables on the left side of the argument operator.

ANS: T

14. In Python there is no restriction on the name of a module file.

ANS: F

15. One of the drawbacks of a modularized program is that the only structure you can use in such a program is the sequence structure.

ANS: F

16. One reason to store graphics functions in a module is so that you can import the module into any program that needs to use those functions.

ANS: T

17. The **randrange** function returns a randomly selected value from a specific sequence of numbers.

ANS: T

18. The **math** function **atan(x)** returns one tangent of **x** in radians.

ANS: F

19. The **math** function **ceil(x)** returns the smallest integer that is greater than or equal to **x**.

ANS: T

20. Unfortunately, there is no way to store and call on functions when using turtle graphics.

ANS: F

**MULTIPLE CHOICE**

1. What is a group of statements that exists within a program for the purpose of performing a specific task?

|  |  |
| --- | --- |
| a. | a function |
| b. | a subtask |
| c. | a process |
| d. | a subprocess |

ANS: A

2. The first line in a function definition is known as the function

|  |  |
| --- | --- |
| a. | header |
| b. | block |
| c. | return |
| d. | parameter |

ANS: A

3. The \_\_\_\_\_\_\_\_\_\_ design technique can be used to break down an algorithm into functions.

|  |  |
| --- | --- |
| a. | subtask |
| b. | block |
| c. | top-down |
| d. | simplification |

ANS: C

4. A set of statements that belong together as a group and contribute to the function definition is known as a

|  |  |
| --- | --- |
| a. | header |
| b. | block |
| c. | return |
| d. | parameter |

ANS: B

5. A(n) \_\_\_\_\_\_\_\_\_\_ chart is also known as a structured chart.

|  |  |
| --- | --- |
| a. | flow |
| b. | data |
| c. | hierarchy |
| d. | organizational |

ANS: C

6. A \_\_\_\_\_\_\_\_\_\_ variable is created inside a function.

|  |  |
| --- | --- |
| a. | global |
| b. | constant |
| c. | named constant |
| d. | local |

ANS: D

7. The \_\_\_\_\_\_\_\_\_\_ of a local variable is the function in which that variable is created.

|  |  |
| --- | --- |
| a. | global reach |
| b. | definition |
| c. | space |
| d. | scope |

ANS: D

8. A(n) \_\_\_\_\_\_\_\_\_\_ is any piece of data that is passed into a function when the function is called.

|  |  |
| --- | --- |
| a. | global variable |
| b. | argument |
| c. | local variable |
| d. | parameter |

ANS: B

9. A(n) \_\_\_\_\_\_\_\_\_\_ is a variable that receives an argument that is passed into a function.

|  |  |
| --- | --- |
| a. | global variable |
| b. | argument |
| c. | named constant |
| d. | parameter |

ANS: D

10. A \_\_\_\_\_\_\_\_\_\_ variable is accessible to all the functions in a program file.

|  |  |
| --- | --- |
| a. | keyword |
| b. | local |
| c. | global |
| d. | string |

ANS: C

11. A \_\_\_\_\_\_\_\_\_\_ constant is a name that references a value that cannot be changed while the program runs.

|  |  |
| --- | --- |
| a. | keyword |
| b. | local |
| c. | global |
| d. | string |

ANS: C

12. When a function is called by its name during the execution of a program, then it is

|  |  |
| --- | --- |
| a. | executed |
| b. | located |
| c. | defined |
| d. | exported |

ANS: A

13. It is recommended that programmers avoid using \_\_\_\_\_\_\_\_\_\_ variables in a program whenever possible.

|  |  |
| --- | --- |
| a. | local |
| b. | global |
| c. | string |
| d. | keyword |

ANS: B

14. The Python library functions that are built into the Python \_\_\_\_\_\_\_\_\_\_ can be used by simply calling the required function.

|  |  |
| --- | --- |
| a. | code |
| b. | compiler |
| c. | linker |
| d. | interpreter |

ANS: D

15. Python comes with \_\_\_\_\_\_\_\_\_\_ functions that have already been prewritten for the programmer.

|  |  |
| --- | --- |
| a. | standard |
| b. | library |
| c. | custom |
| d. | key |

ANS: A

16. What type of function can be used to determine whether a number is even or odd?

|  |  |
| --- | --- |
| a. | even |
| b. | odd |
| c. | math |
| d. | Boolean |

ANS: D

17. A value-returning function is

|  |  |
| --- | --- |
| a. | a single statement that performs a specific task |
| b. | called when you want the function to stop |
| c. | a function that will return a value back to the part of the program that called it |
| d. | a function that receives a value when called |

ANS: C

18. Which of the following statements causes the interpreter to load the contents of the **random** module into memory?

|  |  |
| --- | --- |
| a. | **load random** |
| b. | **import random** |
| c. | **upload random** |
| d. | **download random** |

ANS: B

19. Whic of the following will assign a random integer in the range of **1** through **50** to the variable **number**?

|  |  |
| --- | --- |
| a. | **random(1, 50) = number** |
| b. | **number = random.randint(1, 50)** |
| c. | **randint(1, 50) = number** |
| d. | **number = random(range(1, 50))** |

ANS: B

20. What does the following statement mean?

**num1, num2 = get\_num()**

|  |  |
| --- | --- |
| a. | The function **get\_num()** is expected to return a value for **num1** and for **num2**. |
| b. | The function **get\_num()** is expected to return one value and assign it to **num1** and **num2**. |
| c. | This statement will cause a syntax error. |
| d. | The function **get\_num()** will receive the values stored in **num1** and **num2**. |

ANS: A

21. What will display after the following code is executed?

**def main():**

**print("The answer is", magic(5))**

**def magic(num):**

**answer = num + 2 \* 10**

**return answer**

**main()**

|  |  |
| --- | --- |
| a. | **70** |
| b. | **25** |
| c. | **100** |
| d. | The statement will cause a syntax error. |

ANS: B

22. In a value-returning function, the value of the expression that follows the keyword \_\_\_\_\_\_\_\_\_\_ will be sent back to the part of the program that called the function.

|  |  |
| --- | --- |
| a. | **def** |
| b. | **result** |
| c. | **sent** |
| d. | **return** |

ANS: D

23. The Python standard library's \_\_\_\_\_\_\_\_\_\_ module contains numerous functions that can be used in mathematical calculations.

|  |  |
| --- | --- |
| a. | **math** |
| b. | **string** |
| c. | **random** |
| d. | **number** |

ANS: A

24. Which of the following functions returns the largest integer that is less than or equal to its argument?

|  |  |
| --- | --- |
| a. | **floor** |
| b. | **ceil** |
| c. | **lesser** |
| d. | **greater** |

ANS: A

25. What will be the output after the following code is executed?

**def pass\_it(x, y):**

**z = x + ", " + y**

**return(z)**

**name2 = "Tony"**

**name1 = "Gaddis"**

**fullname = pass\_it(name1, name2)**

**print(fullname)**

|  |  |
| --- | --- |
| a. | **Tony Gaddis** |
| b. | **Gaddis Tony** |
| c. | **Tony, Gaddis** |
| d. | **Gaddis, Tony** |

ANS: D

26. What will be the output after the following code is executed?

**def pass\_it(x, y):**

**z = x , ", " , y**

**num1 = 4**

**num2 = 8**

**answer = pass\_it(num1, num2)**

**print(answer)**

|  |  |
| --- | --- |
| a. | **4, 8** |
| b. | **8, 4** |
| c. | **48** |
| d. | **None** |

ANS: D

27. What will be the output after the following code is executed?

**def pass\_it(x, y):**

**z = y\*\*x**

**return(z)**

**num1 = 3**

**num2 = 4**

**answer = pass\_it(num1, num2)**

**print(answer)**

|  |  |
| --- | --- |
| a. | **81** |
| b. | **64** |
| c. | **12** |
| d. | **None** |

ANS: B

28. What will be displayed after the following code is executed?

**def pass\_it(x, y):**

**z = x\*y**

**result = get\_result(z)**

**return(result)**

**def get\_result(number):**

**z = number + 2**

**return(z)**

**num1 = 3**

**num2 = 4**

**answer = pass\_it(num1, num2)**

**print(answer)**

|  |  |
| --- | --- |
| a. | **12** |
| b. | **9** |
| c. | **14** |
| d. | Nothing, this code contains a syntax error. |

ANS: C

29. What does the following program do?

**import turtle**

**def main():**

**turtle.hideturtle()**

**square(100,0,50,'blue')**

**def square(x, y, width, color):**

**turtle.penup()**

**turtle.goto(x, y)**

**turtle.fillcolor(color)**

**turtle.pendown()**

**turtle.begin\_fill()**

**for count in range(4):**

**turtle.forward(width)**

**turtle.left(90)**

**turtle.end\_fill()**

**main()**

|  |  |
| --- | --- |
| a. | It draws a blue square at coordinates (100, 0), 50 pixels wide, starting at the top right corner. |
| b. | It draws a blue square at coordinates (0, 50), 100 pixels wide, starting at the top right corner. |
| c. | It draws a blue square at coordinates (100, 0), 50 pixels wide, in the lower-left corner. |
| d. | Nothing since you cannot call a function with turtle graphics. |

ANS: C

30. What does the following program do?

**import turtle**

**def main():**

**turtle.hideturtle()**

**square(100,0,50,'blue')**

**def square(x, y, width, color):**

**turtle.penup()**

**turtle.goto(x, y)**

**turtle.fillcolor(color)**

**turtle.pendown()**

**turtle.begin\_fill()**

**for count in range(2):**

**turtle.forward(width)**

**turtle.left(90)**

**turtle.end\_fill()**

**main()**

|  |  |
| --- | --- |
| a. | It draws a blue square. |
| b. | It draws a blue triangle. |
| c. | It draws 2 blue lines. |
| d. | Nothing since you cannot call a function with turtle graphics. |

ANS: B

**COMPLETION**

1. The code for a function is known as a function \_\_\_\_\_\_\_\_\_\_\_.

ANS: definition

2. The function header begins with the keyword \_\_\_\_\_\_\_\_\_\_ and is followed by the name of the function.

ANS: **def**

3. The **main** function contains a program's \_\_\_\_\_\_\_\_\_\_ logic which is the overall logic of the program.

ANS: mainline

4. In a flowchart, a function call is depicted by a(n) \_\_\_\_\_\_\_\_\_\_\_.

ANS: rectangle

5. The top-down design breaks down the overall task of a program into a series of \_\_\_\_\_\_\_\_\_\_.

ANS: subtasks

6. A(n) \_\_\_\_\_\_\_\_\_\_ chart is a visual representation of the relationships between functions.

ANS: hierarchy

7. Arguments are passed by \_\_\_\_\_\_\_\_\_\_ to the corresponding parameter variables in a function.

ANS: position

8. A variable is available only to statements in the variable's \_\_\_\_\_\_\_\_\_\_.

ANS: scope

9. Functions that are in the standard library are stored in files that are known as \_\_\_\_\_\_\_\_\_\_.

ANS: modules

10. To refer to a function in a module, Python uses \_\_\_\_\_\_\_\_\_\_\_ notation.

ANS: dot

11. A value-returning function has a(n) \_\_\_\_\_\_\_\_\_\_\_ statement that sends a value back to the part of the program that called it.

ANS: **return**

12. The \_\_\_\_\_\_\_\_\_\_\_ chart is an effective tool used by programmers to design and document functions.

ANS: IPO

13. The 'P' in the acronym IPO refers to \_\_\_\_\_\_\_\_\_\_\_.

ANS: processing

14. In Python, a module's file name should end in \_\_\_\_\_\_\_\_\_\_\_.

ANS: .py

15. The approach known as \_\_\_\_\_\_\_\_\_\_ makes a program easier to understand, test, and maintain.

ANS: modularization

16. The return values of the trigonometric functions in Python are in \_\_\_\_\_\_\_\_\_\_.

ANS: radians