**Module 8. Functions**

**TRUE/FALSE**

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

Answer: T

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

Answer: T

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

Answer: F

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

Answer: F

1. A function can have default parameter values.

Answer: T

1. The def keyword is used to define a function in Python.

Answer: T

1. Parameters in a function definition must have default values.

Answer: F

1. Functions help in reducing code redundancy.

Answer: T

1. The return statement in a function is optional.

Answer: T

1. A function can return multiple values.

Answer: T

1. Void functions do not return any value.

Answer: T

1. The pass statement is used to end a function definition.

Answer: F

1. You can call a function from within another function.

Answer: T

1. A function must always return a value.

Answer: F

1. The scope of a variable defined inside a function is local to that function.

Answer: T

1. Keyword arguments allow you to pass arguments in any order to a function.

Answer: T

1. The yield statement is used to define a generator function.

Answer: T

1. Function definitions can include nested functions.

Answer: T

1. The \_\_name\_\_ == "\_\_main\_\_" statement is used to ensure code runs only when the module is executed directly.

Answer: T

1. Functions can accept an arbitrary number of positional arguments using \*args.

Answer: T

1. Functions cannot modify global variables.

Answer: F

1. A function can be defined without any parameters.

Answer: T

1. The return statement ends the execution of a function and returns control to the caller.

Answer: T

1. Functions cannot have side effects such as modifying global variables or printing output.

Answer: F

1. Functions help in organizing code into modular units.

Answer: T

**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

Answer: a

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

a. header

b. block

c. return

d. parameter

Answer: a

1. Which 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))

Answer: b

1. 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.

Answer: a

1. What is the purpose of the def keyword in Python?

a. It defines a conditional statement.

b. It defines a function.

c. It defines a loop.

d. It defines a class.

Answer: b

1. Which statement is used to exit a function and return a value?

a. exit

b. quit

c. return

d. end

Answer: c

1. What will be the output of the following code?

*def add(x, y):*

*return x + y*

*print(add(2, 3))*

a. 23

b. 5

c. 6

d. None of the above

Answer: b

1. Which of the following is a valid function call?

a. function\_name

b. function\_name[]

c. function\_name()

d. function\_name{}

Answer: c

1. How do you define a function with no parameters?

a. def function():

b. def function:

c. function()

d. function:

Answer: a

1. What is the result of the following code?

*def multiply(a, b=2):*

*return a \* b*

*print(multiply(3))*

a. 6

b. 9

c. 5

d. Error

Answer: a

1. Which keyword is used to declare a function in Python?

a. func

b. define

c. def

d. function

Answer: c

1. What is the output of the following code?

*def func(x):*

*print(x)*

*func(10)*

a. x

b. func

c. 10

d. None

Answer: c

1. What does the following function return?

*def subtract(a, b):*

*return a – b*

*print(subtract(5, 2))*

a. 7

b. 3

c. 2

d. 1

Answer: b

1. How do you call a function named my\_func with the arguments 2 and 3?

a. my\_func[2, 3]

b. my\_func{2, 3}

c. my\_func(2, 3)

d. my\_func<2, 3>

Answer: c

1. What will be printed by the following code?

*def greet(name="John"):*

*print("Hello, " + name)*

*greet("Alice")*

a. Hello, John

b. Hello, Alice

c. Hello, greet

d. Error

Answer: b

1. Which of the following statements about functions is true?

a. Functions cannot be nested.

b. Functions can be called multiple times.

c. Functions do not improve code readability.

d. Functions must return a value.

Answer: b

1. What is the term for the values passed to a function when it is called?

a. Parameters

b. Arguments

c. Variables

d. Constants

Answer: b

1. How do you specify a function that does not return any value?

a. By not including a return statement.

b. By using void keyword.

c. By setting return type as None.

d. By using pass statement.

Answer: a

1. What will be the output of the following code?

*def divide(a, b):*

*return a / b*

*result = divide(10, 2)*

*print(result)*

a. 5.0

b. 10

c. 2

d. Error

Answer: a

1. Which of the following defines a function with variable-length arguments?

a. def func(\*args):

b. def func(args\*):

c. def func(\*args\*):

d. def func(args):

Answer: a

1. Which statement is used to define a function?

a. function

b. define

c. def

d. fun

Answer: c

1. What is the output of the following code?

*def say\_hello():*

*return "Hello"*

*print(say\_hello())*

a. Hello

b. say\_hello

c. Error

d. None

Answer: a

1. How do you call a function without any arguments?

a. func{}

b. func[]

c. func()

d. func<>

Answer: c

1. What is the term for the code inside a function?

a. Function body

b. Function block

c. Function call

d. Function header

Answer: a

1. What does the following code do?

*def print\_numbers():*

*for i in range(3):*

*print(i)*

*print\_numbers()*

a. Prints 1, 2, 3

b. Prints 0, 1, 2

c. Prints 3, 2, 1

d. Error

Answer: b

1. What is a default parameter?

a. A parameter with a fixed value

b. A parameter that must be provided

c. A parameter with a default value

d. A parameter that cannot be changed

Answer: c

1. Which of the following statements will cause an error?

a. def my\_func(x, y=10):

b. def my\_func(x=10, y):

c. def my\_func(x, y):

d. def my\_func(x=10, y=20):

Answer: b

1. How do you handle an arbitrary number of keyword arguments in a function?

a. Using \*args

b. Using \*\*kwargs

c. Using &args

d. Using &&kwargs

Answer: b

1. What is the purpose of the return statement in a function?

a. To end the function execution

b. To return a value to the caller

c. To specify the function parameters

d. Both a and b

Answer: d

1. What does the yield statement do in a function?

a. It returns a value and ends the function.

b. It creates a generator.

c. It prints a value.

d. It stops the function execution.

Answer: b