PYTHON TEST - 3.3 (DEFINING FUNCTIONS)

Total points 50/50 ?

Defining Functions in Python

STUDENT NAME *

VIVA

✓ **1.** Which keyword is used to define a function in Python? * 1/1

○ a) func

○ b) define

● c) def ✓

○ d) function

✓ **2.** What is the correct syntax to define a function named greet? * 1/1

○ a) function greet():

● b) def greet(): ✓

○ c) func greet():

○ d) greet():

✓ **3.** What will happen if a function is defined but never called? * 1/1

○ a) It will execute automatically

○ b) It will give an error

● c) It won’t execute ✓

○ d) It will print None

✓ **4.** Every function in Python begins with which keyword? * 1/1

○ a) function

● b) def ✓

○ c) lambda

○ d) call

✓ **5.** What is the default return type of a function in Python if not specified? * 1/1

○ a) int

○ b) str

● c) None ✓

○ d) bool

✓ **6.** Which of the following is a valid function definition? * 1/1

● a) def add(x, y): return x + y ✓

○ b) def add(x, y) return x + y

○ c) func add(x, y): return x + y

○ d) define add(x, y): return x + y

✓ **7.** In Python, a function can return: * 1/1

○ a) Only integers

○ b) Only strings

● c) Any data type ✓

○ d) None

✓ **8.** Which keyword is used to exit from a function and return a value? * 1/1

○ a) exit

○ b) end

● c) return ✓

○ d) stop

✓ **9.** A function with no return statement returns: * 1/1

○ a) 0

● b) None ✓

○ c) False

○ d) Empty string

✓ **10.** Can a function return multiple values in Python? * 1/1

○ a) No

○ b) Yes, as a tuple

○ c) Yes, as a list

● d) Both b and c ✓

✓ **11.** Where should the function definition be placed in Python? * 1/1

● a) Before function call ✓

○ b) After function call

○ c) Anywhere in the code

○ d) Inside another function only

✓ **12.** Which of the following is true about defining functions? * 1/1

● a) Function name must start with a letter or underscore ✓

○ b) Function name can start with a number

○ c) Function name can have spaces

○ d) Function name can be any symbol

✓ **13.** What will def func(): pass do? * 1/1

○ a) Raise an error

● b) Create a function that does nothing ✓

○ c) End the program

○ d) Return None automatically

✓ **14.** Which keyword is optional in a function definition? * 1/1

○ a) def

● b) return ✓

○ c) function name

○ d) parentheses

✓ **15.** Which is NOT allowed in a function definition? * 1/1

○ a) Multiple parameters

○ b) No parameters

○ c) Returning multiple values

● d) Function without parentheses ✓

✓ **16.** What is the output of the following? * 1/1

```
def test():
    return
print(test())
```

○ a) 0

● b) None ✓

○ c) Error

○ d) ""

✓ **17.** What will happen if a function has no body? * 1/1

○ a) It will give an error

○ b) It will print None

○ c) It will raise IndentationError

● d) It will do nothing if pass is used ✓

✓ **18.** What will be the output? * 1/1

```
def add(x, y):
    print(x + y)
result = add(2, 3)
print(result)
```

○ a) 5 and 5

● b) 5 and None ✓

○ c) None and 5

○ d) Error

✓ **19.** Which of these can be part of a function definition? * 1/1

○ a) Parameters

○ b) Body

○ c) Return statement

● d) All of the above ✓

✓ **20.** What will be the output? * 1/1

```
def greet():
    return "Hello"
print(greet)
```

○ a) Hello

○ b) Error

● c) Function reference (memory address) ✓

○ d) None

✓ **21.** Can we define a function inside another function in Python? * 1/1

● a) Yes ✓

○ b) No

✓ **22.** Which of the following is correct? * 1/1

○ a) def func[]:

● b) def func(): ✓

○ c) def func:

○ d) def func{}:

✓ **23.** Which keyword defines an anonymous function in Python? * 1/1

○ a) def

○ b) return

● c) lambda ✓

○ d) func

✓ **24.** Which function returns the number of arguments passed in a function call? * 1/1

○ a) count()

○ b) len()

○ c) argc()

● d) None of these ✓

✓ **25.** What is the scope of variables defined inside a function? * 1/1

○ a) Global

● b) Local ✓

○ c) Static

○ d) Constant

✓ **26.** Which is valid? * 1/1

● a) def _func(): pass ✓

○ b) def 1func(): pass

○ c) def func-name(): pass

○ d) def func name(): pass

✓ **27.** Which of these is true? * 1/1

○ a) Function name can be a keyword

● b) Function name cannot be a keyword ✓

✓ **28.** A function defined inside a class is called: * 1/1

○ a) Function

● b) Method ✓

○ c) Constructor

○ d) Destructor

✓ **29.** Which of the following is a correct function definition? * 1/1

● a) def calc(a, b=2): return a+b ✓

○ b) def calc(a=2, b): return a+b

✓ **30.** What happens if return statement is missing? * 1/1

● a) Function returns None ✓

○ b) Function gives error

○ c) Function prints result

○ d) Program stops

✓ **31.** Which keyword defines a generator function? * 1/1

● a) def ✓

○ b) yield

○ c) gen

○ d) return

✓ **32.** A function definition in Python can have: * 1/1

○ a) Only one argument

● b) Unlimited arguments ✓

○ c) Maximum 255 arguments

○ d) No arguments allowed

✓ **33.** Default arguments must appear: * 1/1

○ a) Before non-default arguments

● b) After non-default arguments ✓

○ c) Anywhere

○ d) Not allowed

✓ **34.** Can functions return other functions? * 1/1

● a) Yes ✓

○ b) No

✓ **35.** A recursive function is one that: * 1/1

● a) Calls itself ✓

○ b) Calls another function

○ c) Runs infinitely

○ d) Has no parameters

✓ **36.** What will this return? * 1/1

```
def check():

   return "Hi"

   return "Bye"
print(check())
```

● a) Hi ✓

○ b) Bye

○ c) Both

○ d) Error

✓ **37.** Can a function definition have no parameters? * 1/1

● a) Yes ✓

○ b) No

✓ **38.** Which function shows the docstring of another function? * 1/1

○ a) print()

● b) help() ✓

○ c) info()

○ d) doc()

✓ **39.** A function definition should always have: * 1/1

○ a) def keyword

○ b) Parentheses

○ c) Colon (:) at end of definition line

● d) All of these ✓

✓ **40.** Which keyword is used for documentation in functions? * 1/1

○ a) comment

○ b) doc

● c) """ """ (docstring) ✓

○ d) describe

✓ **41.** Which of the following is valid? * 1/1

● a) def sum(a, b, c=5): return a+b+c ✓

○ b) def sum(a=5, b, c): return a+b+c

✓ **42.** Functions in Python are: * 1/1

● a) First-class objects ✓

○ b) Not objects

○ c) Variables only

○ d) Keywords

✓ **43.** Can functions be passed as arguments? * 1/1

● a) Yes ✓

○ b) No

✓ **44.** Which statement is used to define a function without body temporarily? *1/1

○ a) None

● b) pass ✓

○ c) continue

○ d) break

✓ **45.** Which of these is true for function names? * 1/1

● a) They are case-sensitive ✓

○ b) They are not case-sensitive

✓ **46.** A Python function definition may include: * 1/1

○ a) Parameters

○ b) Default parameters

○ c) Docstring

● d) All of these ✓

✓ **47.** Can a function return different data types at different times? * 1/1

● a) Yes ✓

○ b) No

✓ **48.** What will happen? * 1/1

```
def func():
    print("Hello")
return "Hi"
```

○ a) Prints Hello

○ b) Returns Hi

● c) Error ✓

○ d) None

✓ **49.** Which of the following is NOT true about functions? * 1/1

○ a) Functions help reusability

○ b) Functions reduce code duplication

● c) Functions cannot call themselves ✓

○ d) Functions may have parameters

✓ **50.** Can a function be defined with variable-length arguments? * 1/1

● a) Yes, using *args and **kwargs ✓

○ b) No