### **1. Installation and Setup of Python**

**Q1: Which of the following is NOT an Integrated Development Environment (IDE) used for Python programming?**

a) PyCharm  
b) VSCode  
c) Sublime Text  
d) IntelliJ IDEA

**Answer:** d) IntelliJ IDEA

**Q2: What is the first step to install Python on your system?**

a) Install an IDE  
b) Install a compiler  
c) Download Python from the official website  
d) Configure environmental variables

**Answer:** c) Download Python from the official website

**Q3: Which command can you use to test if Python is correctly installed on your computer?**

a) python test.py  
b) python --version  
c) python check  
d) python info

**Answer:** b) python --version

### **2. Introduction to Python**

**Q4: Who created Python programming language?**

a) Dennis Ritchie  
b) Bjarne Stroustrup  
c) Guido van Rossum  
d) James Gosling

**Answer:** c) Guido van Rossum

**Q5: What is Python primarily used for?**

a) Web development  
b) System programming  
c) Data analysis and machine learning  
d) Game development

**Answer:** c) Data analysis and machine learning

**Q6: Python is an example of which type of programming language?**

a) Compiled language  
b) Scripting language  
c) Machine language  
d) Assembly language

**Answer:** b) Scripting language

### **3. Basic Syntax**

**Q7: What is the correct syntax for printing "Hello, World!" in Python?**

a) echo "Hello, World!"  
b) print("Hello, World!")  
c) printf("Hello, World!")  
d) console.log("Hello, World!")

**Answer:** b) print("Hello, World!")

**Q8: Which of the following is a valid variable name in Python?**

a) 2nd\_variable  
b) variable\_2  
c) #variable  
d) var@iable

**Answer:** b) variable\_2

**Q9: Which Python keyword is used to define a function?**

a) func  
b) def  
c) function  
d) declare

**Answer:** b) def

### **4. Data Types, Variables, and Operators**

**Q10: Which of the following is NOT a basic data type in Python?**

a) Integer  
b) Float  
c) Boolean  
d) Character

**Answer:** d) Character

**Q11: What will be the output of the following Python code?  
x = 5; y = 10; print(x + y)**

a) 15  
b) 5  
c) 10  
d) Error

**Answer:** a) 15

**Q12: Which operator is used to perform division in Python?**

a) //  
b) /  
c) %%  
d) div

**Answer:** b) /

**Q13: What does the == operator do in Python?**

a) Assigns value  
b) Compares two values for equality  
c) Compares two values for inequality  
d) Performs arithmetic addition

**Answer:** b) Compares two values for equality

**Q14: Which of the following is the correct way to define a variable in Python?**

a) int a = 10  
b) 10 = a int  
c) a = 10  
d) var a = 10

**Answer:** c) a = 10

### **5. Input/Output in Python**

**Q15: Which function is used to get input from the user in Python?**

a) output()  
b) input()  
c) scanf()  
d) get\_input()

**Answer:** b) input()

**Q16: What is the output of the following Python code?  
name = input("Enter your name: "); print("Hello, " + name)**

a) Prints "Hello, name"  
b) Asks the user to enter their name and then prints "Hello, name"  
c) Prints an error  
d) Asks the user to enter their name and prints "name"

**Answer:** b) Asks the user to enter their name and then prints "Hello, name"

### **6. Flow of Control: Modules, Branching, If Statements**

**Q17: Which of the following is a valid if statement in Python?**

a) if (x > 5)  
b) if x > 5:  
c) if x > 5 then:  
d) if x > 5 do:

**Answer:** b) if x > 5:

**Q18: What will be the output of the following code?  
x = 10; if x > 5: print("Greater") else: print("Smaller")**

a) Greater  
b) Smaller  
c) Syntax error  
d) None

**Answer:** a) Greater

**Q19: How can you write an else-if condition in Python?**

a) else if  
b) elif  
c) elseif  
d) else: if

**Answer:** b) elif

**Q20: In Python, which statement is used to exit a loop prematurely?**

a) stop  
b) exit  
c) break  
d) return

**Answer:** c) break

**Q21: Which of the following is a valid nested if statement in Python?**

a) if x > 5: if y < 10:  
b) if (x > 5) and (y < 10):  
c) if x > 5: elif y < 10:  
d) if x > 5 { if y < 10 }

**Answer:** a) if x > 5: if y < 10:

### **7. Loops in Python**

**Q22: Which of the following is the correct syntax for a for loop in Python?**

a) for i in range(5):  
b) for (i=0; i<5; i++):  
c) for i = 0; i < 5:  
d) for i in 5:

**Answer:** a) for i in range(5):

**Q23: What will be the output of the following code?  
for i in range(3): print(i)**

a) 0 1 2  
b) 1 2 3  
c) 0 1 2 3  
d) 1 2

**Answer:** a) 0 1 2

**Q24: What is the output of the following code?  
x = 0; while x < 3: print(x); x += 1**

a) 0 1 2  
b) 1 2 3  
c) Infinite loop  
d) Error

**Answer:** a) 0 1 2

**Q25: Which Python keyword is used to skip the current iteration in a loop?**

a) exit  
b) pass  
c) continue  
d) skip

**Answer:** c) continue

**Q26: What will be the output of the following code?  
x = 0; while x < 5: if x == 3: break; print(x); x += 1**

a) 0 1 2 3  
b) 0 1 2  
c) 1 2 3 4  
d) Infinite loop

**Answer:** b) 0 1 2

**Q27: What is the output of the following code?  
for i in range(1, 6, 2): print(i)**

a) 1 2 3 4 5  
b) 1 3 5  
c) 1 2 3 4  
d) 2 4

**Answer:** b) 1 3 5

**Q28: In Python, how do you create an infinite loop?**

a) while True:  
b) while False:  
c) for i in range(0, 0):  
d) loop forever:

**Answer:** a) while True:

**Q29: What does the break statement do in a loop?**

a) Exits the loop and moves to the next statement after the loop  
b) Skips the current iteration and proceeds to the next iteration  
c) Causes an infinite loop  
d) Ends the program execution

**Answer:** a) Exits the loop and moves to the next statement after the loop

### **8. Mixed Questions**

**Q30: What is the correct way to check whether a number is even or odd in Python?**

a) if x % 2 = 0:  
b) if x % 2 == 1:  
c) if x % 2 == 0:  
d) if x % 2 != 0:

**Answer:** c) if x % 2 == 0:

**Q31: What is the output of the following code?  
x = 5; y = 2; print(x / y)**

a) 2.5  
b) 2  
c) 2.0  
d) 5

**Answer:** a) 2.5

**Q32: What does the input() function in Python always return?**

a) Integer  
b) String  
c) Boolean  
d) Float

**Answer:** b) String

**Q33: Which of the following loops is an example of a for loop in Python?**

a) while i < 10:  
b) for i in range(10):  
c) if i < 10:  
d) repeat i < 10:

**Answer:** b) for i in range(10):

**Q34: What is the correct syntax for a while loop in Python?**

a) while (x < 5):  
b) while x < 5:  
c) while x < 5 do:  
d) while x < 5 then:

**Answer:** b) while x < 5:

**Q35: How do you define a function in Python?**

a) function my\_func():  
b) def my\_func:  
c) function my\_func() {}  
d) def my\_func():

**Answer:** d) def my\_func():

**Q36: Which operator is used to assign a value to a variable in Python?**

a) =  
b) ==  
c) :  
d) =>

**Answer:** a) =

**Q37: What will be the output of the following code?  
x = 3; if x == 3: print("Yes"); else: print("No")**

a) Yes  
b) No  
c) Error  
d) None

**Answer:** a) Yes

**Q38: What is the output of the following code?  
x = 7; if x % 2 == 0: print("Even"); else: print("Odd")**

a) Even  
b) Odd  
c) Error  
d) None

**Answer:** b) Odd

**Q39: Which of the following is a correct way to start a comment in Python?**

a) //  
b) #  
c) /\*  
d) --

**Answer:** b) #

**Q40: What is the output of the following Python code?  
x = [1, 2, 3]; print(x[1])**

a) 1  
b) 2  
c) 3  
d) Error

**Answer:** b) 2

**Q41: What will the following code output?  
x = 3; y = 4; z = x + y; print(z)**

a) 34  
b) 7  
c) 12  
d) Error

**Answer:** b) 7

**Q42: Which of the following is a correct comparison operator in Python?**

a) =  
b) ==  
c) <=  
d) All of the above

**Answer:** d) All of the above

**Q43: Which of the following will terminate the loop immediately?**

a) continue  
b) break  
c) pass  
d) exit

**Answer:** b) break

**Q44: What will the following Python code print?  
x = 10; y = 5; print(x == y)**

a) True  
b) False  
c) Error  
d) None

**Answer:** b) False

**Q45: How do you write a comment in Python?**

a) // This is a comment  
b) /\* This is a comment \*/  
c) # This is a comment  
d) comment: This is a comment

**Answer:** c) # This is a comment

**Q46: What will be the result of the expression 5 + 2 \* 3 in Python?**

a) 21  
b) 11  
c) 15  
d) 16

**Answer:** b) 11

**Q47: Which statement in Python is used to define a block of code under a conditional statement?**

a) if  
b) then  
c) elif  
d) Indentation

**Answer:** d) Indentation

**Q48: What is the correct way to call a function in Python?**

a) call func()  
b) func()  
c) func{}  
d) execute func()

**Answer:** b) func()

**Q49: What will this Python code print?  
for i in range(1, 5, 2): print(i)**

a) 1 2 3 4  
b) 1 3  
c) 2 4  
d) 0 1 2 3 4

**Answer:** b) 1 3

**Q50: What will the following code output?  
x = 5; if x < 10: print("Less than 10")**

a) Less than 10  
b) Greater than 10  
c) Error  
d) None

**Answer:** a) Less than 10

**Week 2: Strings, Tuples, and Basic Operations**,

### **1. Pass, Strings, and Tuples**

**Q1: What does the pass statement do in Python?**

a) Exits the current function  
b) Performs no operation, serving as a placeholder  
c) Skips the current iteration of a loop  
d) Creates a new variable

**Answer:** b) Performs no operation, serving as a placeholder

**Q2: Which of the following is NOT a valid way to define a string in Python?**

a) 'Hello, World!'  
b) "Hello, World!"  
c) """Hello, World!"""  
d) <<Hello, World!>>

**Answer:** d) <<Hello, World!>>

**Q3: Which of the following is a tuple in Python?**

a) (1, 2, 3)  
b) [1, 2, 3]  
c) {1, 2, 3}  
d) None

**Answer:** a) (1, 2, 3)

### **2. Accessing Strings**

**Q4: How would you access the first character of a string text = "Python"?**

a) text[1]  
b) text[0]  
c) text[-1]  
d) text[2]

**Answer:** b) text[0]

**Q5: What will the following code print?  
text = "Python"; print(text[1:4])**

a) Pyt  
b) yth  
c) y  
d) thon

**Answer:** b) yth

**Q6: In Python, negative indexing allows you to access elements from the end of the string. What does text[-2] represent for the string text = "Python"?**

a) P  
b) n  
c) o  
d) h

**Answer:** c) o

**Q7: What is the output of the following code?  
text = "Python"; print(text[::2])**

a) Ptn  
b) Pyo  
c) Pytn  
d) t

**Answer:** a) Ptn

**Q8: What does the len() function do when used with a string?**

a) Counts the number of vowels in the string  
b) Returns the length of the string  
c) Returns a list of characters in the string  
d) Converts the string to uppercase

**Answer:** b) Returns the length of the string

### **3. String Methods for Manipulation**

**Q9: Which of the following methods would you use to convert a string to lowercase in Python?**

a) text.toLower()  
b) text.lower()  
c) text.lowercase()  
d) text.downcase()

**Answer:** b) text.lower()

**Q10: What is the output of the following code?  
text = "Python"; print(text.upper())**

a) PYTHON  
b) python  
c) P  
d) Error

**Answer:** a) PYTHON

**Q11: What does the str.strip() method do?**

a) Converts the string to uppercase  
b) Removes leading and trailing spaces from the string  
c) Removes all spaces from the string  
d) Splits the string into a list

**Answer:** b) Removes leading and trailing spaces from the string

**Q12: Which method is used to check if a string starts with a specific prefix?**

a) startswith()  
b) endswith()  
c) contains()  
d) prefix()

**Answer:** a) startswith()

**Q13: Which method will replace all occurrences of a substring within a string?**

a) replace()  
b) substitute()  
c) replace\_all()  
d) sub()

**Answer:** a) replace()

**Q14: What will the following code output?  
text = "Python programming"; print(text.count('o'))**

a) 1  
b) 2  
c) 3  
d) 4

**Answer:** b) 2

### **4. Basic String Operations**

**Q15: Which of the following is the correct syntax to concatenate two strings in Python?**

a) str1 + str2  
b) str1.concat(str2)  
c) concat(str1, str2)  
d) str1, str2

**Answer:** a) str1 + str2

**Q16: What does the \* operator do when used with a string?**

a) Multiplies the string by a given number of times  
b) Concatenates the string with itself  
c) Repeats the string the specified number of times  
d) Raises the string to a power

**Answer:** c) Repeats the string the specified number of times

**Q17: Which of the following is the correct way to format a string in Python?**

a) format("Hello {0}!", name)  
b) "Hello {name}!"  
c) "Hello {}!".format(name)  
d) "Hello %s!" % name

**Answer:** c) "Hello {}!".format(name)

**Q18: What is the output of the following code?  
text = "Python"; print(text \* 2)**

a) PythonPython  
b) Python2  
c) 2Python  
d) Python Python

**Answer:** a) PythonPython

**Q19: What will be the result of this code?  
text = "Python"; print(f"Hello {text}")**

a) Hello Python  
b) Hello {text}  
c) Hello "text"  
d) Hello

**Answer:** a) Hello Python

### **5. Assigning Multiple Values at Once**

**Q21: Which of the following syntax can be used to assign multiple values to multiple variables in one line in Python?**

a) x, y, z = 1, 2, 3  
b) x = y = z = 1  
c) 1 = x = y = z  
d) x, y = 1, 2, 3

**Answer:** a) x, y, z = 1, 2, 3

**Q22: What will the following code output?  
a, b = 5, 10; print(a + b)**

a) 15  
b) 510  
c) 5 10  
d) Error

**Answer:** a) 15

**Q24: What is the output of the following code?  
a, b, c = 1, 2, 3; print(c)**

a) 1  
b) 2  
c) 3  
d) Error

**Answer:** c) 3

**Q25: Which of the following syntax will allow you to assign multiple values to multiple variables on one line?**

a) x, y, z = [1, 2, 3]  
b) x = y = z = [1, 2, 3]  
c) x = 1, y = 2, z = 3  
d) None of the above

**Answer:** a) x, y, z = [1, 2, 3]

### **6. String Slicing**

**Q26: What is the correct syntax to extract a substring from a string text = "Python" starting from index 2 to index 4?**

a) text[2:4]  
b) text[2,4]  
c) text(2,4)  
d) text(2:4)

**Answer:** a) text[2:4]

**Q27: Which method is used to remove leading and trailing whitespace from a string in Python?**

a) strip()  
b) trim()  
c) remove()  
d) clear()

**Answer:** a) strip()

**Q28: What does the following Python code output?  
text = "Hello, World!"; print(text[7:12])**

a) Hello  
b) World  
c) World!  
d) World! (including the space)

**Answer:** b) World

**Q30: How can you get a substring from the beginning of a string text up to index 3?**

a) text[:3]  
b) text[3:]  
c) text(3:)  
d) text[3:3]

**Answer:** a) text[:3]

**Q31: How can you extract every second character from the string text = "Python"?**

a) text[::2]  
b) text[1::2]  
c) text[::3]  
d) text[::4]

**Answer:** a) text[::2]

**Q32: What is the output of the following code?  
text = "Python"; print(text[-3:])**

a) hon  
b) Pyt  
c) ton  
d) Error

**Answer:** a) hon

**Q33: What will the following code print?  
text = "Programming"; print(text[3:7])**

a) ogram  
b) gram  
c) ramm  
d) Program

**Answer:** b) gram

**Q34: How can you extract a substring from index 2 to the end of a string text = "Python"?**

a) text[2:]  
b) text[:2]  
c) text[2:-1]  
d) text[-2:]

**Answer:** a) text[2:]

**Q35: How would you extract the first three characters from the string text = "Python"?**

a) text[:3]  
b) text[3:]  
c) text[1:3]  
d) text[3]

**Answer:** a) text[:3]

**Q36: How can you access the last character of a string text = "Python" using negative indexing?**

a) text[-1]  
b) text[-2]  
c) text[0]  
d) text[-3]

**Answer:** a) text[-1]

### **7. Mixed Questions**

**Q37: What does the following code output?  
text = "Python"; print(text[1:])**

a) thon  
b) ython  
c) P  
d) Error

**Answer:** b) ython

**Q38: What will be the result of this code?  
text = "Python"; print(text[::-1])**

a) Python  
b) nohtyP  
c) t  
d) thon

**Answer:** b) nohtyP

**Q39: How would you assign multiple variables in one line?**

a) x, y = 5, 10  
b) x = y = 5  
c) x 5, y 10  
d) x = 5, y = 10

**Answer:** a) x, y = 5, 10

**Q40: What will be the output of this code?  
text = "Programming"; print(text[:5] + text[6:])**

a) Programming  
b) Proamming  
c) ogramming  
d) Progrming

**Answer:** a) Programming

**Q41: What is the purpose of the join() method in Python?**

a) Joins two strings  
b) Combines a list of strings into a single string  
c) Removes whitespace from the string  
d) Divides a string into a list

**Answer:** b) Combines a list of strings into a single string

**Q42: What will be the output of this code?  
text = "Python"; print(text[0:3])**

a) Py  
b) Pyt  
c) P  
d) Python

**Answer:** b) Pyt

**Q43: How would you create a tuple with multiple values in Python?**

a) x = (1, 2, 3)  
b) x = [1, 2, 3]  
c) x = {1, 2, 3}  
d) x = "1, 2, 3"

**Answer:** a) x = (1, 2, 3)

**Q44: How can you check if a string contains a specific substring in Python?**

a) if 'Python' in text:  
b) if text.contains('Python'):  
c) if 'Python' == text:  
d) if text.include('Python'):

**Answer:** a) if 'Python' in text:

**Q45: Which of these methods can be used to check if a string is a digit in Python?**

a) isdigit()  
b) isnumber()  
c) isnumeric()  
d) digit()

**Answer:** a) isdigit()

**Q46: Which method can be used to convert a string to a list of words?**

a) split()  
b) list()  
c) wordsplit()  
d) convert()

**Answer:** a) split()

**Q47: What does the startswith() method do?**

a) Checks if the string starts with a specific substring  
b) Checks if the string ends with a specific substring  
c) Converts the string to lowercase  
d) Trims the string

**Answer:** a) Checks if the string starts with a specific substring

**Q48: How can you check if a string is empty?**

a) if text == ""  
b) if len(text) == 0  
c) if not text  
d) All of the above

**Answer:** d) All of the above

**Q49: What will be the output of this code?  
text = "Python"; print(text[1:5:2])**

a) Py  
b) Pth  
c) ytn  
d) y

**Answer:** b) Pth

**Q50: What does the str.format() method do?**

a) Formats the string by replacing placeholders with values  
b) Converts the string to uppercase  
c) Returns a substring  
d) Checks if the string is empty

**Answer:** a) Formats the string by replacing placeholders with values

**Week 3: Working with Dictionaries**,

**Q32: What is the output of this code?  
my\_dict = {'a': 1, 'b': 2}; my\_dict['b'] = 3; my\_dict['c'] = 4; print(my\_dict)**

a) {'a': 1, 'b': 3, 'c': 4}  
b) {'a': 1, 'b': 2, 'c': 4}  
c) {'b': 3, 'c': 4, 'a': 1}  
d) Error

**Answer:** a) {'a': 1, 'b': 3, 'c': 4}

**Q33: Which of the following will remove the last item in a dictionary?**

a) dict.delete()  
b) dict.pop()  
c) dict.popitem()  
d) dict.remove()

**Answer:** c) dict.popitem()

**Q34: What does the items() method return in a dictionary?**

a) A list of values in the dictionary  
b) A list of keys in the dictionary  
c) A list of key-value pairs in the dictionary  
d) A set of keys and values

**Answer:** c) A list of key-value pairs in the dictionary

**Q35: How do you convert a dictionary to a list of tuples containing the key-value pairs?**

a) list(dict)  
b) dict.keys()  
c) dict.items()  
d) dict.to\_list()

**Answer:** c) dict.items()

**Q36: What will be the result of the following code?  
my\_dict = {'x': 10, 'y': 20, 'z': 30}; my\_dict['a'] = 40; print(len(my\_dict))**

a) 4  
b) 3  
c) 5  
d) 10

**Answer:** a) 4

**Q37: What does the get() method do when the key is not found in a dictionary?**

a) Returns None  
b) Raises an error  
c) Returns the value 'Not Found' (if specified)  
d) Deletes the key-value pair

**Answer:** c) Returns the value 'Not Found' (if specified)

**Q38: Which method will return all the values of a dictionary?**

a) dict.values()  
b) dict.get\_values()  
c) dict.all\_values()  
d) dict.list()

**Answer:** a) dict.values()

**Q39: What is the purpose of the setdefault() method in a dictionary?**

a) To add a default key-value pair if the key does not exist  
b) To remove all default values  
c) To get the value of a key and set it if not already set  
d) To update an existing key-value pair

**Answer:** a) To add a default key-value pair if the key does not exist

**Q40: How would you combine two dictionaries dict1 and dict2?**

a) dict1 + dict2  
b) dict1.update(dict2)  
c) dict1.append(dict2)  
d) dict1.merge(dict2)

**Answer:** b) dict1.update(dict2)

**Q41: Which of the following will remove all keys and values from the dictionary?**

a) del dict  
b) dict.clear()  
c) dict.remove()  
d) dict.reset()

**Answer:** b) dict.clear()

**Q42: What will be the result of the following code?  
my\_dict = {'apple': 1, 'banana': 2, 'cherry': 3}; del my\_dict['banana']; print(my\_dict)**

a) {'apple': 1, 'cherry': 3}  
b) {'banana': 2, 'apple': 1, 'cherry': 3}  
c) {'apple': 1, 'banana': 2}  
d) Error

**Answer:** a) {'apple': 1, 'cherry': 3}

**Q43: Which method will check if a specific key exists in a dictionary?**

a) dict.has\_key()  
b) dict.keyexists()  
c) dict.contains()  
d) key in dict

**Answer:** d) key in dict

**Q44: What will be the result of the following code?  
my\_dict = {'a': 1, 'b': 2}; print('a' in my\_dict)**

a) True  
b) False  
c) 1  
d) None

**Answer:** a) True

**Q45: How would you access the value associated with the key 'b' in the dictionary my\_dict = {'a': 1, 'b': 2} using the get() method?**

a) my\_dict.get('b')  
b) my\_dict.get('b', None)  
c) my\_dict['b']  
d) All of the above

**Answer:** d) All of the above

**Q46: What is the return type of the keys() method in a dictionary?**

a) A list of keys  
b) A set of keys  
c) A tuple of keys  
d) A dictionary with keys only

**Answer:** b) A set of keys

**Q47: What does the pop() method do in a dictionary?**

a) Removes the first item  
b) Removes and returns the value associated with the specified key  
c) Removes the last item  
d) Clears the dictionary

**Answer:** b) Removes and returns the value associated with the specified key

**Q48: Which of the following will you use to get all the key-value pairs from a dictionary as tuples?**

a) dict.items()  
b) dict.keys()  
c) dict.values()  
d) dict.pairs()

**Answer:** a) dict.items()

**Q50: Which of the following statements will raise an error?**

a)   
my\_dict['a'] = 1b) del my\_dict['b']  
c) my\_dict.clear()  
d) my\_dict['c']

**Answer:** d) my\_dict['c'] (if 'c' is not already a key in the dictionary)

**LIST**

### **Q1: What is the main characteristic of a Python list?**

a) Lists are immutable.  
b) Lists can store elements of only one data type.  
c) Lists are ordered collections of items.  
d) Lists are only for numeric data.

**Answer:** c) Lists are ordered collections of items.

### **Q2: How do you create an empty list in Python?**

a) list = []  
b) list = {}  
c) list = ()  
d) list = ""

**Answer:** a) list = []

### **Q3: Which of the following is a valid list in Python?**

a) list = [1, 2, 3, 'Python']  
b) list = (1, 2, 3, 'Python')  
c) list = {1, 2, 3, 'Python'}  
d) list = 1, 2, 3, 'Python'

**Answer:** a) list = [1, 2, 3, 'Python']

### **\*\*Q4: What will be the result of the following code?**

my\_list = [1, 2, 3, 4]; print(my\_list[2])\*\*

a) 1  
b) 2  
c) 3  
d) 4

**Answer:** c) 3

### **Q5: Which of the following will retrieve the last element of a list my\_list = [10, 20, 30, 40]?**

a) my\_list[-1]  
b) my\_list[0]  
c) my\_list[1]  
d) my\_list[4]

**Answer:** a) my\_list[-1]

### **\*\*Q6: What will be the result of the following code?**

my\_list = [1, 2, 3, 4]; my\_list[1] = 5; print(my\_list)\*\*

a) [1, 5, 3, 4]  
b) [1, 2, 3, 5]  
c) [5, 2, 3, 4]  
d) Error

**Answer:** a) [1, 5, 3, 4]

### **Q7: How do you add an element 10 to the end of a list my\_list = [1, 2, 3]?**

a) my\_list.append(10)  
b) my\_list.add(10)  
c) my\_list.insert(10)  
d) my\_list.extend(10)

**Answer:** a) my\_list.append(10)

### **Q8: Which method is used to insert an element at a specific index in a list?**

a) append()  
b) insert()  
c) extend()  
d) add()

**Answer:** b) insert()

### **Q9: How do you add multiple elements [4, 5, 6] to the list my\_list = [1, 2, 3]?**

a) my\_list.insert([4, 5, 6])  
b) my\_list.append([4, 5, 6])  
c) my\_list.extend([4, 5, 6])  
d) my\_list.add([4, 5, 6])

**Answer:** c) my\_list.extend([4, 5, 6])

### **\*\*Q10: What will be the output of the following code?**

my\_list = [1, 2, 3]; my\_list.remove(2); print(my\_list)\*\*

a) [1, 3]  
b) [1, 2]  
c) [3]  
d) [2, 3]

**Answer:** a) [1, 3]

### **Q11: Which method removes the last element from a list?**

a) remove()  
b) pop()  
c) clear()  
d) delete()

**Answer:** b) pop()

### **\*\*Q12: What will the following code output?**

my\_list = [1, 2, 3, 4, 5]; my\_list.pop(2); print(my\_list)\*\*

a) [1, 2, 4, 5]  
b) [1, 2, 3, 4]  
c) [2, 3, 4, 5]  
d) [1, 2, 3, 5]

**Answer:** a) [1, 2, 4, 5]

### **Q13: What is the result of my\_list.clear() for my\_list = [1, 2, 3, 4]?**

a) Removes all elements from the list  
b) Removes only the first element  
c) Removes the last element  
d) Clears the last two elements

**Answer:** a) Removes all elements from the list

### **\*\*Q14: What is the result of the following code?**

my\_list = [10, 20, 30]; print(20 in my\_list)\*\*

a) True  
b) False  
c) 20  
d) None

**Answer:** a) True

### **Q15: What is the output of my\_list = [10, 20, 30]; print(len(my\_list))?**

a) 3  
b) 4  
c) 10  
d) None

**Answer:** a) 3

### **Q16: How would you repeat the list [1, 2] three times?**

a) [1, 2] \* 3  
b) [1, 2] + 3  
c) [1, 2].repeat(3)  
d) repeat([1, 2], 3)

**Answer:** a) [1, 2] \* 3

### **Q17: What is the result of concatenating two lists list1 = [1, 2] and list2 = [3, 4]?**

a) [1, 2, 3, 4]  
b) [1, 2] + [3, 4]  
c) [1, 2].append([3, 4])  
d) Error

**Answer:** a) [1, 2, 3, 4]

### **Q18: Which of the following methods returns a list of indexes of all occurrences of an element?**

a) find()  
b) index()  
c) count()  
d) index\_of()

**Answer:** b) index()

### **Q19: What will the result of my\_list = [1, 2, 3]; my\_list[1:3] = [4, 5] be?**

a) [1, 4, 5]  
b) [1, 4, 5, 3]  
c) [4, 5, 3]  
d) [4, 5]

**Answer:** a) [1, 4, 5]

### **Q20: Which of the following methods can be used to get a sublist from an existing list?**

a) split()  
b) slice()  
c) sublist()  
d) list()

**Answer:** b) slice()

### **Q21: Which of the following operators can be used to repeat a list multiple times?**

a) \*  
b) +  
c) =  
d) \*\*

**Answer:** a) \*

### **\*\*Q22: What will the following code output?**

my\_list = ['a', 'b', 'c']; my\_list += ['d', 'e']; print(my\_list)\*\*

a) ['a', 'b', 'c', 'd', 'e']  
b) ['a', 'b', 'c']  
c) ['d', 'e']  
d) Error

**Answer:** a) ['a', 'b', 'c', 'd', 'e']

### **Q23: How would you access the first element of a list my\_list = [10, 20, 30, 40]?**

a) my\_list[1]  
b) my\_list[0]  
c) my\_list.first()  
d) my\_list[10]

**Answer:** b) my\_list[0]

### **Q24: Which of the following functions can be used to map a function to each item in a list?**

a) map()  
b) apply()  
c) filter()  
d) loop()

**Answer:** a) map()

### **Q25: How do you join a list of strings ['Hello', 'World'] into a single string with a space?**

a) ' '.join(['Hello', 'World'])  
b) 'Hello' + ' ' + 'World'  
c) join(['Hello', 'World'])  
d) '.join('Hello World')

**Answer:** a) ' '.join(['Hello', 'World'])

### **Q26: What does the count() method do in a list?**

a) Returns the number of elements in a list  
b) Returns the index of the first occurrence of an element  
c) Counts the number of occurrences of a specific element  
d) Returns a boolean indicating if a list is empty

**Answer:** c) Counts the number of occurrences of a specific element

### **\*\*Q27: What is the result of the following code?**

my\_list = [1, 2, 3]; my\_list.pop(); print(my\_list)\*\*

a) [1, 2]  
b) [2, 3]  
c) [1, 3]  
d) Error

**Answer:** a) [1, 2]

### **Q28: How do you create a new list by applying a function to each item in an existing list?**

a) apply()  
b) map()  
c) list()  
d) function()

**Answer:** b) map()

### **Q29: What is the correct way to delete the element at index 2 in a list my\_list = [10, 20, 30, 40]?**

a) my\_list.pop(2)  
b) my\_list.delete(2)  
c) my\_list.remove(2)  
d) my\_list[2] = None

**Answer:** a) my\_list.pop(2)

### **\*\*Q30: What is the result of the following code?**

my\_list = [1, 2, 3, 4, 5]; print(my\_list[::2])\*\*

a) [1, 3, 5]  
b) [1, 2, 3]  
c) [2, 4]  
d) [5, 4, 3, 2, 1]

**Answer:** a) [1, 3, 5]

### **Q31: Which operator can be used to concatenate two lists?**

a) +  
b) \*  
c) &  
d) #

**Answer:** a) +

### **\*\*Q32: What will the following code output?**

my\_list = [1, 2, 3]; my\_list = my\_list \* 2; print(my\_list)\*\*

a) [1, 2, 3, 1, 2, 3]  
b) [1, 2, 3]  
c) [2, 3, 4]  
d) None

**Answer:** a) [1, 2, 3, 1, 2, 3]

### **Q33: Which of the following is NOT a valid way to initialize an empty list?**

a) []  
b) list()  
c) () d) None

**Answer:** c) ()

### **Q34: What is the result of my\_list = [1, 2, 3]; my\_list[::-1]?**

a) [3, 2, 1]  
b) [1, 2, 3]  
c) [2, 3, 1]  
d) Error

**Answer:** a) [3, 2, 1]

### **Q35: What does the reverse() method do in a list?**

a) Reverses the elements of the list in place  
b) Reverses the list and returns it  
c) Swaps the first and last element  
d) Sorts the list in reverse order

**Answer:** a) Reverses the elements of the list in place

### **Q36: How can you create a list with 100 elements, all initialized to zero?**

a) [0] \* 100  
b) list(0, 100)  
c) [100]  
d) repeat(0, 100)

**Answer:** a) [0] \* 100

### **Q37: What does the index() method do in a list?**

a) Returns the position of the first occurrence of an element  
b) Returns the total number of elements  
c) Reverses the list  
d) Removes duplicates

**Answer:** a) Returns the position of the first occurrence of an element

### **Q38: Which of the following methods can be used to sort a list in Python?**

a) sort()  
b) order()  
c) sorted()  
d) Both a and c

**Answer:** d) Both a and c

### **\*\*Q39: What is the result of the following code?**

my\_list = [10, 20, 30, 40]; my\_list = my\_list[1:3]; print(my\_list)\*\*

a) [20, 30]  
b) [10, 20]  
c) [30, 40]  
d) Error

**Answer:** a) [20, 30]

### **Q40: How do you delete the last element of the list my\_list = [1, 2, 3, 4]?**

a) my\_list.remove()  
b) my\_list.pop()  
c) my\_list.delete()  
d) my\_list.remove(4)

**Answer:** b) my\_list.pop()

### **Q41: How do you combine two lists list1 = [1, 2] and list2 = [3, 4]?**

a) list1 + list2  
b) list1.append(list2)  
c) list1.insert(list2)  
d) list1.extend(list2)

**Answer:** a) list1 + list2

### **Q42: What does the map() function do in the context of a list?**

a) Adds new elements to a list  
b) Reverses the list  
c) Applies a function to every element of a list  
d) Filters the list

**Answer:** c) Applies a function to every element of a list

### **Q43: How can you get the maximum value in a list my\_list = [1, 2, 3, 4]?**

a) max(my\_list)  
b) min(my\_list)  
c) my\_list.max()  
d) max(my\_list())

**Answer:** a) max(my\_list)

### **Q44: How can you get the sum of elements in a list my\_list = [1, 2, 3, 4]?**

a) sum(my\_list)  
b) my\_list.sum()  
c) total(my\_list)  
d) add(my\_list)

**Answer:** a) sum(my\_list)

### **Q45: Which function is used to generate a list from a string using delimiters?**

a) split()  
b) join()  
c) map()  
d) splitlines()

**Answer:** a) split()

### **\*\*Q46: What will the following code output?**

my\_list = [10, 20, 30]; my\_list.append(40); print(my\_list)\*\*

a) [10, 20, 30, 40]  
b) [10, 20, 40]  
c) [40]  
d) [10, 30, 20, 40]

**Answer:** a) [10, 20, 30, 40]

### **Q47: What is the time complexity of accessing an element by index in a list?**

a) O(n)  
b) O(log n)  
c) O(1)  
d) O(n log n)

**Answer:** c) O(1)

### **Q48: What is the difference between remove() and pop() in lists?**

a) remove() removes by index, pop() removes by value.  
b) remove() removes by value, pop() removes by index.  
c) remove() can only remove the first element.  
d) There is no difference.

**Answer:** b) remove() removes by value, pop() removes by index.

### **\*\*Q49: What is the result of the following code?**

my\_list = [1, 2, 3]; my\_list.append([4, 5]); print(my\_list)\*\*

a) [1, 2, 3, [4, 5]]  
b) [1, 2, 3, 4, 5]  
c) [4, 5, 1, 2, 3]  
d) Error

**Answer:** a) [1, 2, 3, [4, 5]]

### **Q50: How can you convert a string s = "hello" to a list of characters?**

a) list(s)  
b) split(s)  
c) map(s)  
d) join(s)

**Answer:** a) list(s)

Here are **50 MCQs** covering **Week 5: Functions and Methods** for Sessions 5 & 6:

### **Q1: What is the correct syntax to define a function in Python?**

a) function my\_function():  
b) def my\_function:  
c) function def my\_function():  
d) def my\_function():

**Answer:** d) def my\_function():

### **Q2: Which of the following is used to call a function in Python?**

a) function call my\_function()  
b) call function my\_function()  
c) my\_function()  
d) execute my\_function()

**Answer:** c) my\_function()

### **\*\*Q3: What will the following code output?**

def greet(name): print("Hello", name); greet("Alice")\*\*

a) Hello  
b) Alice  
c) Hello Alice  
d) Error

**Answer:** c) Hello Alice

### **Q4: What is the default return value of a function if no return statement is specified?**

a) None  
b) 0  
c) False  
d) Empty string

**Answer:** a) None

### **Q5: Which of the following types of functions is used for simple one-liner functions?**

a) Recursive function  
b) Lambda function  
c) Normal function  
d) Built-in function

**Answer:** b) Lambda function

### **Q6: What is the correct way to define a lambda function that adds two numbers?**

a) lambda add(x, y): return x + y  
b) lambda x, y: x + y  
c) lambda(x, y): x + y  
d) lambda: x + y

**Answer:** b) lambda x, y: x + y

### **Q7: Which of the following is an example of a recursive function?**

a) A function that calls itself  
b) A function that runs a loop  
c) A function that uses an if statement  
d) A function that accepts parameters

**Answer:** a) A function that calls itself

### **Q8: What is the main difference between positional and keyword arguments?**

a) Positional arguments are passed by name, keyword arguments by position  
b) Positional arguments are passed by position, keyword arguments by name  
c) Keyword arguments must be declared before positional arguments  
d) Positional arguments cannot have default values

**Answer:** b) Positional arguments are passed by position, keyword arguments by name

### **Q9: Which of the following is the correct syntax to call a function with positional arguments?**

a) function(param1=5, param2=10)  
b) function(5, 10)  
c) function(param1=5, 10)  
d) function(5, param2=10)

**Answer:** b) function(5, 10)

### **Q10: What will happen if you call a function with missing required positional arguments?**

a) The function will throw a TypeError  
b) The function will automatically set the argument to None  
c) The function will return 0  
d) The function will run without any issues

**Answer:** a) The function will throw a TypeError

### **Q11: Which of the following is a correct way to define a function with a default argument value?**

a) def my\_function(x=10):  
b) def my\_function(x=10, y):  
c) def my\_function(x, y=20):  
d) def my\_function(x=10, y=20=30):

**Answer:** a) def my\_function(x=10):

### **Q12: How can you pass multiple arguments to a function without specifying each argument explicitly?**

a) Use the \*args syntax  
b) Use \*\*kwargs  
c) Use \*args and \*\*kwargs  
d) All of the above

**Answer:** d) All of the above

### **Q13: What is the difference between \*args and \*\*kwargs in function definitions?**

a) \*args accepts multiple positional arguments, and \*\*kwargs accepts multiple keyword arguments  
b) \*args accepts keyword arguments, and \*\*kwargs accepts positional arguments  
c) \*args is used for default values  
d) There is no difference

**Answer:** a) \*args accepts multiple positional arguments, and \*\*kwargs accepts multiple keyword arguments

### **\*\*Q14: What will the following code output?**

def my\_func(\*args): print(args); my\_func(1, 2, 3)\*\*

a) 1, 2, 3  
b) (1, 2, 3)  
c) [1, 2, 3]  
d) Error

**Answer:** b) (1, 2, 3)

### **Q15: Which function argument type allows the caller to pass a variable number of keyword arguments?**

a) \*args  
b) \*\*kwargs  
c) \*varargs  
d) \*arguments

**Answer:** b) \*\*kwargs

### **Q16: What will happen if you use a global variable inside a function without declaring it as global?**

a) The global variable will be accessible without any issue  
b) Python will create a new local variable with the same name  
c) It will cause a syntax error  
d) It will raise a NameError

**Answer:** b) Python will create a new local variable with the same name

### **Q17: What is the purpose of the global keyword in a function?**

a) To define a global variable  
b) To access a global variable inside a function  
c) To prevent a function from accessing the global scope  
d) To declare variables as constants

**Answer:** b) To access a global variable inside a function

### **Q18: What is the scope of a variable defined inside a function?**

a) Local scope  
b) Global scope  
c) Both local and global scopes  
d) No scope

**Answer:** a) Local scope

### **Q19: How would you access the type of a variable in Python?**

a) type(variable)  
b) variable.type()  
c) type(variable())  
d) gettype(variable)

**Answer:** a) type(variable)

### **Q20: Which function returns a string that represents the properties of an object in Python?**

a) str()  
b) repr()  
c) format()  
d) list()

**Answer:** b) repr()

### **Q21: What is the correct syntax for defining a function that takes two arguments and returns their sum?**

a) def sum(x, y): return x + y  
b) def sum(x, y): return x - y  
c) def sum(x, y) { return x + y }  
d) sum(x, y) = x + y

**Answer:** a) def sum(x, y): return x + y

### **Q22: How can you print the attributes and methods of an object in Python?**

a) dir(object)  
b) print(object)  
c) object.methods()  
d) attributes(object)

**Answer:** a) dir(object)

### **Q23: What does the help() function do in Python?**

a) Returns a description of the argument object  
b) Provides the type of the object  
c) Helps to debug the code  
d) Returns the version of Python

**Answer:** a) Returns a description of the argument object

### **Q24: Which of the following functions can be used to check the data type of a variable in Python?**

a) type()  
b) data\_type()  
c) typeOf()  
d) isType()

**Answer:** a) type()

### **Q25: Which of the following is the correct syntax to define a function that returns the product of two numbers?**

a) def product(x, y): return x \* y  
b) def product(x, y) { return x \* y }  
c) def product(x, y): print(x \* y)  
d) def product(x, y): x \* y

**Answer:** a) def product(x, y): return x \* y

### **\*\*Q26: What will the following code output?**

def add(x, y=5): return x + y; print(add(10))\*\*

a) 15  
b) 10  
c) None  
d) Error

**Answer:** a) 15

### **Q27: Which of the following methods is used to convert a list into a string in Python?**

a) str()  
b) join()  
c) concatenate()  
d) to\_string()

**Answer:** b) join()

### **Q28: What is the purpose of the dir() function in Python?**

a) To list the attributes and methods of an object  
b) To return the current directory  
c) To check if an attribute exists  
d) To list all variables in the current scope

**Answer:** a) To list the attributes and methods of an object

### **Q29: What does the round() function do in Python?**

a) Returns the floor value of a number  
b) Returns the ceiling value of a number  
c) Rounds a number to a specified decimal place  
d) Rounds a number to the nearest integer

**Answer:** c) Rounds a number to a specified decimal place

### **Q30: Which of the following is the correct way to define a recursive function to compute the factorial of a number?**

a) def factorial(n): return n \* factorial(n-1)  
b) def factorial(n): if n == 1: return 1; else: return n \* factorial(n-1)  
c) def factorial(n): return n \* n  
d) def factorial(n): if n == 1: return n; else: return 1

**Answer:** b) def factorial(n): if n == 1: return 1; else: return n \* factorial(n-1)

### **Q31: What will happen if you try to access a local variable outside its scope?**

a) It will return the default value  
b) It will raise a NameError  
c) It will return None  
d) It will print the variable value

**Answer:** b) It will raise a NameError

### **Q32: How do you handle variable scope in Python?**

a) By using global variables  
b) By using local() and global() functions  
c) By using global and nonlocal keywords  
d) By passing variables as arguments

**Answer:** c) By using global and nonlocal keywords

### **Q33: Which Python function is used to return the length of an object?**

a) length()  
b) len()  
c) size()  
d) count()

**Answer:** b) len()

### **\*\*Q34: What will the following code output?**

def show\_message(message="Hello"): print(message); show\_message()\*\*

a) Hello  
b) message  
c) None  
d) Error

**Answer:** a) Hello

### **\*\*Q35: What will the following code output?**

def func(x): return x\*\*2; print(func(5))\*\*

a) 25  
b) 50  
c) 5  
d) Error

**Answer:** a) 25

### **Q36: Which built-in function can you use to get the memory address of an object?**

a) id()  
b) mem()  
c) memory()  
d) address()

**Answer:** a) id()

### **Q37: What does the sorted() function do in Python?**

a) Sorts the object in reverse order  
b) Sorts an object and returns a new sorted list  
c) Sorts an object in place  
d) Returns the largest item in the object

**Answer:** b) Sorts an object and returns a new sorted list

### **\*\*Q38: What is the output of the following code?**

def power(x, y=2): return x \*\* y; print(power(3))\*\*

a) 9  
b) 8  
c) 6  
d) None

**Answer:** a) 9

### **Q39: Which built-in function in Python can be used to calculate the absolute value of a number?**

a) abs()  
b) absolute()  
c) value()  
d) absval()

**Answer:** a) abs()

### **Q40: How do you define a function that accepts an arbitrary number of arguments in Python?**

a) By using \*args in the function definition  
b) By using \*\*kwargs in the function definition  
c) By passing a list of arguments  
d) By defining multiple parameters

**Answer:** a) By using \*args in the function definition

### **\*\*Q41: What is the output of the following code?**

def multiply(x, y=2): return x \* y; print(multiply(4, 5))\*\*

a) 20  
b) 10  
c) 8  
d) Error

**Answer:** a) 20

### **Q42: Which of the following is NOT a built-in function in Python?**

a) print()  
b) input()  
c) sqrt()  
d) type()

**Answer:** c) sqrt()

### **Q43: How do you define a function with multiple default arguments?**

a) def function(x=1, y=2):  
b) def function(x=1; y=2):  
c) def function(x=1, y):  
d) def function(x, y=2):

**Answer:** a) def function(x=1, y=2):

### **Q44: Which built-in function returns the type of an object?**

a) type()  
b) str()  
c) id()  
d) isinstance()

**Answer:** a) type()

### **Q45: Which of the following is true for lambda functions?**

a) They can have multiple expressions  
b) They always return None  
c) They are limited to a single expression  
d) They cannot accept arguments

**Answer:** c) They are limited to a single expression

### **Q46: How can you define a function that accepts keyword arguments only?**

a) By using \*args  
b) By using \*\*kwargs  
c) By using both \*args and \*\*kwargs  
d) By defining only default arguments

**Answer:** b) By using \*\*kwargs

### **Q47: How can you print the docstring of a function?**

a) help(function)  
b) doc(function)  
c) print(function)  
d) describe(function)

**Answer:** a) help(function)

### **Q48: What will happen if you use a variable outside the function where it is defined without declaring it global?**

a) It will cause a NameError  
b) Python will use the global variable  
c) Python will create a new local variable  
d) The variable will return None

**Answer:** a) It will cause a NameError

### **\*\*Q49: What will be the output of the following code?**

def greet(): return "Hello!"; print(greet())\*\*

a) Hello!  
b) greet()  
c) None  
d) Error

**Answer:** a) Hello!

### **Q50: What is the return value of a lambda function?**

a) A string  
b) An integer  
c) The value returned by the expression  
d) None

**Answer:** c) The value returned by the expression

These **50 MCQs** cover various aspects of **functions and methods** in Python, including function definitions, types, arguments, variable scopes, and built-in functions.

**TUPLE**

### **Q1: What is the key difference between a tuple and a list in Python?**

a) Tuples are mutable, lists are immutable  
b) Lists are ordered, tuples are unordered  
c) Tuples are immutable, lists are mutable  
d) Tuples cannot store numbers, but lists can

**Answer:** c) Tuples are immutable, lists are mutable

### **Q2: Which of the following is the correct way to define a tuple in Python?**

a) tuple = [1, 2, 3]  
b) tuple = {1, 2, 3}  
c) tuple = (1, 2, 3)  
d) tuple = 1, 2, 3

**Answer:** c) tuple = (1, 2, 3)

### **Q3: How do you define an empty tuple in Python?**

a) empty\_tuple = []  
b) empty\_tuple = {}  
c) empty\_tuple = ()  
d) empty\_tuple = None

**Answer:** c) empty\_tuple = ()

### **Q4: Which of the following methods can you use to access an element of a tuple?**

a) Indexing  
b) Slicing  
c) Both indexing and slicing  
d) Neither indexing nor slicing

**Answer:** c) Both indexing and slicing

### **\*\*Q5: What will the following code output?**

tuple = (10, 20, 30); print(tuple[1])\*\*

a) 10  
b) 20  
c) 30  
d) Error

**Answer:** b) 20

### **Q6: How can you retrieve the last element of a tuple t = (5, 10, 15, 20)?**

a) t[0]  
b) t[3]  
c) t[-1]  
d) t[last]

**Answer:** c) t[-1]

### **\*\*Q7: What will the following code output?**

tuple = (1, 2, 3, 4, 5); print(tuple[1:4])\*\*

a) (1, 2, 3)  
b) (2, 3, 4)  
c) (2, 3, 4, 5)  
d) (1, 2, 3, 4)

**Answer:** b) (2, 3, 4)

### **Q8: How can you create a tuple with one element?**

a) tuple = (1)  
b) tuple = [1]  
c) tuple = (1,)  
d) tuple = {1}

**Answer:** c) tuple = (1,)

### **Q9: Which of the following is a valid operation on tuples?**

a) Concatenation  
b) Repetition  
c) Membership tests  
d) All of the above

**Answer:** d) All of the above

### **\*\*Q10: What will the following code output?**

tuple = (1, 2, 3); print(1 in tuple)\*\*

a) True  
b) False  
c) Error  
d) None

**Answer:** a) True

### **Q11: Can you modify the elements of a tuple after it is created?**

a) Yes, because tuples are mutable  
b) No, because tuples are immutable  
c) Yes, using slicing  
d) No, but you can create a new tuple

**Answer:** b) No, because tuples are immutable

### **\*\*Q12: What is the output of the following code?**

tuple = (10, 20, 30, 40); print(tuple[1:])\*\*

a) (20, 30, 40)  
b) (10, 20)  
c) (20, 30, 40, 50)  
d) (30, 40)

**Answer:** a) (20, 30, 40)

### **Q13: How can you concatenate two tuples t1 = (1, 2) and t2 = (3, 4)?**

a) t1.add(t2)  
b) t1 + t2  
c) t1.concat(t2)  
d) t1.append(t2)

**Answer:** b) t1 + t2

### **Q14: Which of the following will raise an error when executed?**

a) tuple = (1, 2, 3); tuple[0] = 10  
b) tuple = (1, 2, 3); tuple[1] = 20  
c) tuple = (1, 2, 3); tuple[0:2] = (10, 20)  
d) tuple = (1, 2, 3); del tuple[1]

**Answer:** a) tuple = (1, 2, 3); tuple[0] = 10

### **Q15: Which method can be used to find the length of a tuple?**

a) len()  
b) length()  
c) count()  
d) size()

**Answer:** a) len()

### **Q16: How can you unpack a tuple into separate variables?**

a) x, y, z = (1, 2, 3)  
b) x, y = (1, 2, 3)  
c) x = (1, 2, 3)  
d) x = (1, 2)

**Answer:** a) x, y, z = (1, 2, 3)

### **Q17: What happens when you try to unpack a tuple into more variables than there are elements?**

a) It will raise a ValueError  
b) It will assign None to the extra variables  
c) It will repeat the last element of the tuple  
d) It will return 0

**Answer:** a) It will raise a ValueError

### **\*\*Q18: What is the output of the following code?**

tuple = (5, 10, 15); x, y, z = tuple; print(y)\*\*

a) 5  
b) 10  
c) 15  
d) None

**Answer:** b) 10

### **Q19: Which of the following is NOT a valid operation for tuples?**

a) Concatenation  
b) Repetition  
c) Item assignment  
d) Membership test

**Answer:** c) Item assignment

### **\*\*Q20: What is the output of the following code?**

tuple = (1, 2, 3, 4, 5); print(tuple[-3:])\*\*

a) (3, 4, 5)  
b) (2, 3, 4)  
c) (1, 2, 3)  
d) (4, 5)

**Answer:** a) (3, 4, 5)

### **Q21: Which of the following operations can be performed on a tuple?**

a) Addition  
b) Deletion  
c) Modification  
d) Sorting

**Answer:** a) Addition

### **Q22: What does the count() method return when applied to a tuple?**

a) The number of elements in the tuple  
b) The number of occurrences of a given element  
c) The first element in the tuple  
d) The index of the element

**Answer:** b) The number of occurrences of a given element

### **Q23: Which of the following is the correct way to create a tuple with only one element?**

a) tuple = (1)  
b) tuple = [1]  
c) tuple = (1,)  
d) tuple = (1, 1)

**Answer:** c) tuple = (1,)

### **\*\*Q24: What will the following code output?**

tuple = (1, 2, 3, 4, 5); print(tuple[::-1])\*\*

a) (1, 2, 3, 4, 5)  
b) (5, 4, 3, 2, 1)  
c) (2, 3, 4, 5)  
d) Error

**Answer:** b) (5, 4, 3, 2, 1)

### **Q25: Which of the following will result in a tuple containing a single element 3?**

a) tuple = (3)  
b) tuple = [3]  
c) tuple = (3,)  
d) tuple = (3, 3)

**Answer:** c) tuple = (3,)

### **Q26: Which of the following is the correct syntax to create a tuple from a list?**

a) tuple = list(t)  
b) tuple = tuple(t)  
c) tuple = convert(t)  
d) tuple = [t]

**Answer:** b) tuple = tuple(t)

### **Q27: What does the index() method return when applied to a tuple?**

a) The last element of the tuple  
b) The index of the specified element  
c) The first element of the tuple  
d) A tuple of all indexes

**Answer:** b) The index of the specified element

### **Q28: What will happen when you try to modify an element of a tuple?**

a) It will change the value  
b) It will throw a TypeError  
c) It will create a new tuple  
d) It will work fine without any errors

**Answer:** b) It will throw a TypeError

### **Q29: Which of the following methods will allow you to count how many times an element appears in a tuple?**

a) tuple.count()  
b) tuple.size()  
c) tuple.length()  
d) tuple.index()

**Answer:** a) tuple.count()

### **Q30: What is the correct syntax to delete an entire tuple?**

a) del tuple  
b) tuple.clear()  
c) del tuple[]  
d) tuple.remove()

**Answer:** a) del tuple

### **Q31: Which of the following is NOT allowed for tuples?**

a) Indexing  
b) Slicing  
c) Assignment  
d) Concatenation

**Answer:** c) Assignment

### **Q32: Which of the following will result in a TypeError when applied to a tuple?**

a) tuple + tuple  
b) tuple \* 2  
c) tuple[0] = 10  
d) tuple[::-1]

**Answer:** c) tuple[0] = 10

### **Q33: Can a tuple contain lists as its elements?**

a) Yes, tuples can contain any type of object, including lists  
b) No, tuples can only contain other tuples  
c) Yes, but only immutable types can be inside a tuple  
d) No, tuples cannot contain lists

**Answer:** a) Yes, tuples can contain any type of object, including lists

### **Q34: What is the method to check if an element exists in a tuple?**

a) contains()  
b) exists()  
c) in  
d) find()

**Answer:** c) in

### **\*\*Q35: What will the following code output?**

tuple = (1, 2, 3); print(tuple[0:2])\*\*

a) (1, 2)  
b) (2, 3)  
c) (3, 4)  
d) Error

**Answer:** a) (1, 2)

### **Q36: Which of the following operations is NOT allowed on tuples?**

a) Tuple assignment  
b) Tuple concatenation  
c) Tuple multiplication  
d) Tuple slicing

**Answer:** a) Tuple assignment

### **Q37: What is the maximum number of elements a tuple can hold?**

a) There is no limit  
b) 1  
c) 10  
d) 100

**Answer:** a) There is no limit

### **Q38: Which of the following is true about tuple unpacking?**

a) Tuples cannot be unpacked  
b) Unpacking works for both lists and tuples  
c) Only one element of a tuple can be unpacked at a time  
d) Unpacking is only allowed for dictionaries

**Answer:** b) Unpacking works for both lists and tuples

### **Q39: How do you check the number of elements in a tuple?**

a) len(tuple)  
b) count(tuple)  
c) size(tuple)  
d) elements(tuple)

**Answer:** a) len(tuple)

### **Q40: Which method would you use to find the position of a specific element in a tuple?**

a) index()  
b) position()  
c) find()  
d) locate()

**Answer:** a) index()

### **Q41: Can a tuple be used as a key in a dictionary?**

a) Yes, since tuples are immutable  
b) No, because tuples are mutable  
c) Yes, but only tuples with strings  
d) No, because tuples cannot contain multiple elements

**Answer:** a) Yes, since tuples are immutable

### **\*\*Q42: What is the result of the following code?**

tuple = (10, 20, 30, 40); print(tuple[-2:])\*\*

a) (30, 40)  
b) (10, 20)  
c) (20, 30, 40)  
d) Error

**Answer:** a) (30, 40)

### **Q43: How do you check if a tuple contains an element?**

a) Use if element in tuple  
b) Use tuple.contains()  
c) Use tuple.exists()  
d) Use tuple.index()

**Answer:** a) Use if element in tuple

### **Q44: How can you store multiple tuples in a single tuple?**

a) By appending them  
b) By creating nested tuples  
c) By converting to a list first  
d) By using extend()

**Answer:** b) By creating nested tuples

### **\*\*Q45: What will the following code output?**

tuple = (1, 2, 3); tuple = tuple + (4, 5); print(tuple)\*\*

a) (1, 2, 3, 4, 5)  
b) (1, 2, 3)  
c) (4, 5)  
d) Error

**Answer:** a) (1, 2, 3, 4, 5)

### **Q46: What does tuple unpacking allow you to do?**

a) Assign multiple values from a tuple to variables  
b) Merge two tuples  
c) Modify elements in a tuple  
d) Add new elements to a tuple

**Answer:** a) Assign multiple values from a tuple to variables

### **Q47: What is the method to remove all elements from a tuple?**

a) tuple.remove()  
b) tuple.clear()  
c) del tuple  
d) tuple.delete()

**Answer:** c) del tuple

### **Q48: What is the return value of the index() method if the element is not found in the tuple?**

a) None  
b) Error  
c) -1  
d) 0

**Answer:** b) Error

### **Q49: What happens if you concatenate a tuple with a list?**

a) It will raise a TypeError  
b) It will concatenate the elements  
c) It will merge the two structures  
d) It will modify the list

**Answer:** a) It will raise a TypeError

### **Q50: How can you initialize a tuple containing repeated values?**

a) tuple = (5, 5, 5)  
b) tuple = 5 \* 3  
c) tuple = (5,) \* 3  
d) tuple = (5, 5)

**Answer:** c) tuple = (5,) \* 3

These **50 MCQs** cover the different aspects of working with **tuples** in Python, including their definition, operations, unpacking, and various methods.

**Here are 50 MCQs covering Week 7: File Handling in Python (Session 9):**

### **Q1: Which of the following functions is used to open a file in Python?**

a) open()  
b) file()  
c) load()  
d) read()

**Answer:** a) open()

### **Q2: What does the open() function return in Python?**

a) A file object  
b) The file contents  
c) The file name  
d) A list of file names

**Answer:** a) A file object

### **Q3: Which of the following file modes is used for opening a file in write mode?**

a) r  
b) w  
c) a  
d) rb

**Answer:** b) w

### **Q4: Which of the following file modes is used for opening a file in append mode?**

a) r  
b) w  
c) a  
d) rw

**Answer:** c) a

### **Q5: What will happen if you open a file in write mode (w) and the file does not exist?**

a) It will raise an error  
b) The file will be created automatically  
c) It will create a file with default content  
d) The program will freeze

**Answer:** b) The file will be created automatically

### **Q6: What does the close() function do when applied to a file?**

a) It saves the file content  
b) It releases the file resources  
c) It deletes the file  
d) It writes the content of the file to disk

**Answer:** b) It releases the file resources

### **Q7: Which of the following methods is used to read the entire content of a file in Python?**

a) read()  
b) readline()  
c) readlines()  
d) close()

**Answer:** a) read()

### **Q8: Which of the following methods reads one line at a time from a file?**

a) read()  
b) readline()  
c) readlines()  
d) open()

**Answer:** b) readline()

### **Q9: What is the correct way to write to a file in Python?**

a) write()  
b) writeline()  
c) add()  
d) insert()

**Answer:** a) write()

### **Q10: What does the writelines() method do?**

a) Writes a single string to a file  
b) Writes multiple lines to a file  
c) Writes a string and automatically adds new lines  
d) Writes and saves the file content

**Answer:** b) Writes multiple lines to a file

### **Q11: What is the purpose of using the os module for file handling?**

a) To perform file I/O operations  
b) To navigate directories and paths  
c) To handle file content formatting  
d) To encrypt files

**Answer:** b) To navigate directories and paths

### **Q12: Which method from the os module is used to check if a file exists?**

a) os.check()  
b) os.isfile()  
c) os.exists()  
d) os.path.isfile()

**Answer:** d) os.path.isfile()

### **Q13: Which of the following methods can be used to delete a file in Python?**

a) os.remove()  
b) os.delete()  
c) os.remove\_file()  
d) os.delete\_file()

**Answer:** a) os.remove()

### **Q14: Which function from the shutil module is used to copy a file?**

a) shutil.copyfile()  
b) shutil.move()  
c) shutil.remove()  
d) shutil.copy()

**Answer:** a) shutil.copyfile()

### **Q15: Which of the following is used to handle directory operations in Python?**

a) os  
b) os.path  
c) shutil  
d) All of the above

**Answer:** d) All of the above

### **Q16: How can you get the current working directory in Python?**

a) os.getcwd()  
b) os.current()  
c) os.getcwd()  
d) os.path.current()

**Answer:** c) os.getcwd()

### **Q17: What function from the os module is used to create a directory?**

a) os.mkdir()  
b) os.create()  
c) os.newdir()  
d) os.make()

**Answer:** a) os.mkdir()

### **Q18: What will happen if you try to open a file in read mode (r) and the file does not exist?**

a) The program will create a new file  
b) It will raise an error  
c) It will create an empty file  
d) The program will hang indefinitely

**Answer:** b) It will raise an error

### **Q19: Which of the following is used to read all lines from a file into a list?**

a) readlines()  
b) read()  
c) linebyline()  
d) getlines()

**Answer:** a) readlines()

### **Q20: Which of the following methods can be used to rename a file in Python?**

a) os.rename()  
b) os.rename\_file()  
c) os.move()  
d) os.replace()

**Answer:** a) os.rename()

### **Q21: What function in the pathlib module is used to get the absolute path of a file?**

a) pathlib.absolute()  
b) pathlib.resolve()  
c) pathlib.abspath()  
d) pathlib.fullpath()

**Answer:** b) pathlib.resolve()

### **Q22: Which of the following file modes allows both reading and writing to a file?**

a) r+  
b) w+  
c) rw  
d) a+

**Answer:** a) r+

### **Q23: What will happen if you try to open a file in write mode (w) that is already opened in read mode (r)?**

a) The file will be opened in read-write mode  
b) It will throw an error  
c) The write operation will overwrite the content of the file  
d) The file will be locked

**Answer:** b) It will throw an error

### **Q24: Which of the following operations can be performed using the pathlib module?**

a) Checking if a file exists  
b) Navigating directories  
c) Getting the file size  
d) All of the above

**Answer:** d) All of the above

### **Q25: How can you check if a directory exists using the os module?**

a) os.exists()  
b) os.path.exists()  
c) os.path.isdir()  
d) os.isdir()

**Answer:** c) os.path.isdir()

### **Q26: What is the result of calling os.path.join('folder', 'file.txt')?**

a) folder/file.txt  
b) folder\file.txt  
c) folder\file  
d) folder/file

**Answer:** a) folder/file.txt

### **Q27: Which of the following is true about file handling in Python?**

a) A file can only be opened in one mode at a time  
b) File modes allow both reading and writing at the same time  
c) File handling is not supported in Python  
d) File handling only supports text files

**Answer:** a) A file can only be opened in one mode at a time

### **Q28: Which Python library provides file path manipulation and directory operations?**

a) shutil  
b) os  
c) pathlib  
d) os.path

**Answer:** c) pathlib

### **Q29: What is the default mode for opening a file in Python using open()?**

a) w  
b) r  
c) a  
d) rb

**Answer:** b) r

### **Q30: Which of the following is used to copy an entire directory in Python?**

a) shutil.copy()  
b) shutil.copytree()  
c) shutil.copydir()  
d) shutil.copymove()

**Answer:** b) shutil.copytree()

### **Q31: What does the os.path.abspath() method do?**

a) It returns the absolute path of the current working directory  
b) It returns the relative path of a file  
c) It returns the absolute path of a file  
d) It returns the last modified path

**Answer:** c) It returns the absolute path of a file

### **Q32: What method would you use to check if a path is a directory in Python?**

a) os.isdir()  
b) os.path.isdir()  
c) os.directory()  
d) os.path.isdir()

**Answer:** b) os.path.isdir()

### **Q33: What happens when you open a file in binary mode (rb)?**

a) It opens the file as a regular text file  
b) It opens the file as a binary file for reading  
c) It converts the content to text  
d) It raises an error

**Answer:** b) It opens the file as a binary file for reading

### **Q34: How can you copy a file in Python using the shutil module?**

a) shutil.copy()  
b) shutil.move()  
c) shutil.copyfile()  
d) shutil.copyfileobj()

**Answer:** c) shutil.copyfile()

### **Q35: What does the os.rmdir() method do?**

a) Removes a file  
b) Removes a directory  
c) Removes a folder and its contents  
d) Removes a file and its contents

**Answer:** b) Removes a directory

### **Q36: Which of the following file handling methods is used to append to a file?**

a) write()  
b) writelines()  
c) append()  
d) a+

**Answer:** d) a+

### **Q37: What happens when you open a file in 'rb' mode and try to write to it?**

a) The file is opened for reading and writing  
b) It raises an error as binary files cannot be written  
c) It will overwrite the file content  
d) It appends content to the file

**Answer:** b) It raises an error as binary files cannot be written

### **Q38: How can you rename a directory in Python?**

a) os.rename()  
b) os.move()  
c) os.rename\_dir()  
d) os.path.rename()

**Answer:** a) os.rename()

### **Q39: Which of the following methods is used to check the existence of a file in Python?**

a) os.path.check()  
b) os.exists()  
c) os.path.exists()  
d) os.isfile()

**Answer:** c) os.path.exists()

### **Q40: Which of the following functions is used to remove an empty directory?**

a) os.remove()  
b) os.remove\_dir()  
c) os.rmdir()  
d) shutil.rmdir()

**Answer:** c) os.rmdir()

### **Q41: What is the purpose of using the pathlib module?**

a) For managing file permissions  
b) For manipulating file and directory paths  
c) For reading files only  
d) For editing files

**Answer:** b) For manipulating file and directory paths

### **Q42: What happens if you attempt to open a file for writing but the file is open elsewhere in read-only mode?**

a) The file cannot be opened  
b) It overwrites the file  
c) It appends content to the file  
d) It raises an error

**Answer:** a) The file cannot be opened

### **Q43: What will the os.path.join() function do when given two arguments ('folder', 'file.txt')?**

a) Join them into a single string with no separator  
b) Combine them with a system-specific separator  
c) Join them into a list  
d) Return an error

**Answer:** b) Combine them with a system-specific separator

### **Q44: What is the output of shutil.copytree('dir1', 'dir2')?**

a) It copies the contents of 'dir1' to 'dir2'  
b) It creates a copy of 'dir1' in 'dir2'  
c) It moves 'dir1' to 'dir2'  
d) It raises an error if 'dir2' already exists

**Answer:** b) It creates a copy of 'dir1' in 'dir2'

### **Q45: Which of the following is used to list all files and directories in a specified path?**

a) os.list()  
b) os.files()  
c) os.listdir()  
d) pathlib.listdir()

**Answer:** c) os.listdir()

### **Q46: What does the os.path.splitext() method do?**

a) Splits the file into multiple parts  
b) Extracts the file name and extension  
c) Removes the extension from the file name  
d) Combines the file name and extension

**Answer:** b) Extracts the file name and extension

### **Q47: What is the file path separator used by the os module in Windows?**

a) /  
b) \  
c) :  
d) |

**Answer:** b) \

### **Q48: How can you check the permissions of a file in Python?**

a) Use os.permissions()  
b) Use os.check\_permissions()  
c) Use os.access()  
d) Use file.permissions()

**Answer:** c) Use os.access()

### **Q49: What is the role of the shutil module in Python?**

a) It manages file permissions  
b) It handles file and directory operations like copying and moving  
c) It handles encryption and compression  
d) It manipulates file contents

**Answer:** b) It handles file and directory operations like copying and moving

### **Q50: Which function is used to get the file size in Python?**

a) os.getsize()  
b) os.path.size()  
c) os.path.getsize()  
d) os.size()

**Answer:** c) os.path.getsize()

**Week 8: Advanced Python Concepts - Object-Oriented Programming**

### **Q1: What is Object-Oriented Programming (OOP)?**

a) A method of programming using functions and variables  
b) A programming paradigm that uses objects and classes  
c) A technique for parallel programming  
d) A tool for handling errors in Python

**Answer:** b) A programming paradigm that uses objects and classes

### **Q2: Which of the following is NOT one of the core principles of OOP?**

a) Encapsulation  
b) Inheritance  
c) Polymorphism  
d) Multithreading

**Answer:** d) Multithreading

### **Q3: What is encapsulation in OOP?**

a) The ability to create new methods in subclasses  
b) Hiding the internal state of an object and restricting direct access  
c) A way to represent real-world entities in code  
d) Reusing methods and attributes in multiple classes

**Answer:** b) Hiding the internal state of an object and restricting direct access

### **Q4: Which of the following OOP principles allows a child class to inherit methods from a parent class?**

a) Encapsulation  
b) Inheritance  
c) Polymorphism  
d) Abstraction

**Answer:** b) Inheritance

### **Q5: What does polymorphism in OOP allow?**

a) Defining methods with the same name in multiple classes  
b) Creating abstract methods in classes  
c) Hiding the internal state of an object  
d) Creating objects with different names

**Answer:** a) Defining methods with the same name in multiple classes

### **Q6: What is abstraction in OOP?**

a) Hiding unnecessary details and showing only relevant information  
b) Reusing code from different classes  
c) Defining methods that can be called by other objects  
d) Creating child classes

**Answer:** a) Hiding unnecessary details and showing only relevant information

### **Q7: Which of the following keyword is used to define a class in Python?**

a) object  
b) def  
c) class  
d) init

**Answer:** c) class

### **Q8: How do you create an object in Python?**

a) Using the create keyword  
b) Using the new keyword  
c) By calling the class name as if it were a function  
d) Using the object() function

**Answer:** c) By calling the class name as if it were a function

### **Q9: What is the purpose of the \_\_init\_\_ method in a Python class?**

a) To define default values for the object  
b) To initialize the class itself  
c) To define attributes of the class  
d) To initialize the object when it is created

**Answer:** d) To initialize the object when it is created

### **Q10: In Python, how can you call a method of a class?**

a) Using self.method()  
b) Using class.method()  
c) Using object.method()  
d) Both b and c

**Answer:** d) Both b and c

### **Q11: Which of the following is an example of inheritance in Python?**

a) Creating a child class that inherits methods from a parent class  
b) Defining private variables in a class  
c) Using abstract classes in Python  
d) Defining a method that cannot be overridden

**Answer:** a) Creating a child class that inherits methods from a parent class

### **Q12: What is method overloading in Python?**

a) Defining multiple methods with the same name in a class  
b) Creating multiple constructors for a class  
c) Changing the number of arguments for the same method name  
d) Overriding methods in child classes

**Answer:** c) Changing the number of arguments for the same method name

### **Q13: Which method is used to inherit functionality from a parent class in Python?**

a) super()  
b) inherit()  
c) extend()  
d) class()

**Answer:** a) super()

### **Q14: What does method overriding allow in Python?**

a) To call a method of the parent class  
b) To change the behavior of a method in the child class  
c) To add new methods to the class  
d) To delete methods from the parent class

**Answer:** b) To change the behavior of a method in the child class

### **Q15: Which of the following is true about private members in Python?**

a) Private members cannot be accessed outside the class  
b) Private members are always public  
c) Private members can only be used inside the class  
d) Private members are declared using the private keyword

**Answer:** a) Private members cannot be accessed outside the class

### **Q16: How are private attributes in a class defined in Python?**

a) By adding a single underscore before the attribute name  
b) By adding two underscores before the attribute name  
c) By using the private keyword  
d) Private attributes do not exist in Python

**Answer:** b) By adding two underscores before the attribute name

### **Q17: What is the purpose of the re module in Python?**

a) To work with binary data  
b) To manage system files  
c) To handle regular expressions  
d) To perform networking operations

**Answer:** c) To handle regular expressions

### **Q18: Which of the following is a correct syntax for creating a regular expression object in Python?**

a) re.create(r'\d+')  
b) re.compile(r'\d+')  
c) re.match(r'\d+')  
d) re.re()

**Answer:** b) re.compile(r'\d+')

### **Q19: Which Python method is used to search for a pattern in a string using a regular expression?**

a) re.search()  
b) re.match()  
c) re.find()  
d) re.scan()

**Answer:** a) re.search()

### **Q20: Which of the following is the correct usage of re.match() in Python?**

a) It matches a pattern from the start of the string  
b) It matches a pattern anywhere in the string  
c) It matches the pattern at the end of the string  
d) It does not use any pattern

**Answer:** a) It matches a pattern from the start of the string

### **Q21: What is the result of calling re.findall(r'\d+', '123 abc 456 def')?**

a) ['123', '456']  
b) ['123', ' abc', ' 456', ' def']  
c) ['123 abc', '456 def']  
d) ['abc', 'def']

**Answer:** a) ['123', '456']

### **Q22: What is the purpose of the re.sub() function?**

a) To search for a pattern in the string  
b) To substitute a pattern with a new string  
c) To compile a pattern  
d) To match a pattern

**Answer:** b) To substitute a pattern with a new string

### **Q23: Which of the following Python class methods are automatically called when an object is created?**

a) \_\_str\_\_()  
b) \_\_init\_\_()  
c) \_\_del\_\_()  
d) \_\_new\_\_()

**Answer:** b) \_\_init\_\_()

### **Q24: In Python, what does the @staticmethod decorator do?**

a) It defines a method that can be accessed without an instance  
b) It defines a class method  
c) It is used to define an instance method  
d) It allows the method to be overridden

**Answer:** a) It defines a method that can be accessed without an instance

### **Q25: Which of the following is used to define a class variable in Python?**

a) self.variable  
b) cls.variable  
c) object.variable  
d) @variable

**Answer:** b) cls.variable

### **Q26: How do you prevent a method from being overridden in Python?**

a) By using the final keyword  
b) By using the static keyword  
c) By using the @method decorator  
d) There is no direct way to prevent method overriding in Python

**Answer:** d) There is no direct way to prevent method overriding in Python

### **Q27: How do you define a class attribute in Python?**

a) Using cls  
b) Using self  
c) Using class  
d) Using @ decorator

**Answer:** a) Using cls

### **Q28: Which of the following methods is used to delete an attribute in Python?**

a) delattr()  
b) delete()  
c) remove()  
d) discard()

**Answer:** a) delattr()

### **Q29: How do you define an abstract method in Python?**

a) Using the abstract keyword  
b) Using the def keyword  
c) Using the @abstractmethod decorator  
d) Abstract methods are not supported in Python

**Answer:** c) Using the @abstractmethod decorator

### **Q30: What does super() do in Python?**

a) It refers to the parent class and allows access to its methods  
b) It creates a new instance of the parent class  
c) It defines a new class method  
d) It adds a method to the parent class

**Answer:** a) It refers to the parent class and allows access to its methods

### **Q31: What is a class method in Python?**

a) A method that is called on an instance of the class  
b) A method that is called on the class itself rather than the instance  
c) A method that defines instance attributes  
d) A method that creates an object

**Answer:** b) A method that is called on the class itself rather than the instance

### **Q32: Which method is used to represent an object as a string in Python?**

a) \_\_print\_\_()  
b) \_\_str\_\_()  
c) \_\_repr\_\_()  
d) \_\_string\_\_()

**Answer:** b) \_\_str\_\_()

### **Q33: How can you create an instance of a class in Python?**

a) By calling the class with parentheses  
b) By using the new() function  
c) By using the object() function  
d) By using the instance() function

**Answer:** a) By calling the class with parentheses

### **Q34: What is method resolution order (MRO) in Python?**

a) The order in which methods are invoked  
b) The order in which classes are inherited  
c) The order in which methods are added to a class  
d) The order in which methods are printed

**Answer:** a) The order in which methods are invoked

### **Q35: What does re.match(r'\d+', 'abc123') return?**

a) None  
b) '123'  
c) 'abc123'  
d) 'abc'

**Answer:** a) None

### **Q36: How do you check if a regular expression pattern exists in a string?**

a) re.test()  
b) re.match()  
c) re.search()  
d) re.find()

**Answer:** c) re.search()

### **Q37: Which of the following is an example of polymorphism in Python?**

a) A class that has multiple methods with the same name  
b) A method that changes its behavior based on input  
c) Inheriting from multiple classes  
d) Using super() to access parent class methods

**Answer:** b) A method that changes its behavior based on input

### **Q38: What will the following Python code output?**

python

Copy code

class A:

def \_\_init\_\_(self):

self.name = "Class A"

class B(A):

def \_\_init\_\_(self):

super().\_\_init\_\_()

obj = B()

print(obj.name)

a) Class A  
b) Class B  
c) AttributeError  
d) None

**Answer:** a) Class A

### **Q39: What is a benefit of using encapsulation in OOP?**

a) Improved performance  
b) Preventing the unauthorized access of object data  
c) Faster object creation  
d) Decreased complexity

**Answer:** b) Preventing the unauthorized access of object data

### **Q40: What does the re.findall() method return in Python?**

a) A list of matched strings  
b) The first matched string  
c) A string containing the matched text  
d) None

**Answer:** a) A list of matched strings

### **Q41: Which of the following is a correct example of a private attribute in Python?**

a) \_\_name  
b) \_name  
c) name  
d) private\_name

**Answer:** a) \_\_name

### **Q42: What is the correct way to create a regular expression for matching any digit?**

a) r'[0-9]'  
b) r'[A-Z]'  
c) r'[a-z]'  
d) r'\w'

**Answer:** a) r'[0-9]'

### **Q43: Which Python module is used to work with regular expressions?**

a) re  
b) regex  
c) os  
d) pattern

**Answer:** a) re

### **Q44: How can you define an abstract class in Python?**

a) Using the abstract keyword  
b) Using the @abstractmethod decorator  
c) Using the class keyword only  
d) Python does not support abstract classes

**Answer:** b) Using the @abstractmethod decorator

### **Q45: What does the following Python code do?**

python

Copy code

import re

pattern = re.compile(r'\d+')

result = pattern.findall('123 abc 456 def')

print(result)

a) Prints the first number found in the string  
b) Prints the entire string  
c) Prints a list of numbers found in the string  
d) Raises an error

**Answer:** c) Prints a list of numbers found in the string

### **Q46: In which case would you use method overloading?**

a) When you want to change the method signature based on input  
b) When you want to override a method in the child class  
c) When you need to define a method for each object  
d) When you need to call a method multiple times

**Answer:** a) When you want to change the method signature based on input

### **Q47: How can you prevent method overriding in Python?**

a) Python doesn't allow method overriding  
b) By using the final keyword  
c) By using the @nooverride decorator  
d) There is no way to prevent method overriding directly

**Answer:** d) There is no way to prevent method overriding directly

### **Q48: What does re.sub(r'\d+', 'X', '123 abc 456') return?**

a) '123 abc 456'  
b) 'X abc X'  
c) 'abc X'  
d) 'X abc X def'

**Answer:** b) 'X abc X'

### **Q49: Which of the following is NOT a feature of OOP in Python?**

a) Polymorphism  
b) Inheritance  
c) Dynamic Typing  
d) Automatic Garbage Collection

**Answer:** c) Dynamic Typing

### **Q50: Which method is used to remove the first occurrence of a value in a list?**

a) remove()  
b) pop()  
c) clear()  
d) delete()

**Answer:** a) remove()

**Here are 50 MCQs covering Week 9: Exception Handling**

### **Q1: What is an exception in Python?**

a) A type of function  
b) An error that occurs during the execution of a program  
c) A data type  
d) A method used to catch errors

**Answer:** b) An error that occurs during the execution of a program

### **Q2: Which of the following is a common built-in exception in Python?**

a) FileNotFoundError  
b) FileError  
c) Error  
d) ExceptionError

**Answer:** a) FileNotFoundError

### **Q3: What happens when an exception occurs in Python?**

a) The program stops running immediately without any feedback  
b) The program terminates without catching the error  
c) The exception is raised and Python looks for an exception handler  
d) Python ignores the exception and continues execution

**Answer:** c) The exception is raised and Python looks for an exception handler

### **Q4: What is the purpose of the try block in Python exception handling?**

a) To catch errors  
b) To execute code that might cause exceptions  
c) To define the function signature  
d) To create a custom exception

**Answer:** b) To execute code that might cause exceptions

### **Q5: What does the except block do in Python?**

a) It defines a function  
b) It executes code when no exception is raised  
c) It is used to specify what happens when an exception is caught  
d) It ends the program execution

**Answer:** c) It is used to specify what happens when an exception is caught

### **Q6: How do you handle multiple exceptions in one except clause?**

a) Using multiple except blocks  
b) By chaining multiple try blocks  
c) Using parentheses to separate exceptions  
d) Using a finally block

**Answer:** c) Using parentheses to separate exceptions

### **Q7: What is the finally block used for in Python exception handling?**

a) To raise an exception  
b) To ensure that a block of code is always executed, regardless of exceptions  
c) To log the exception  
d) To end the program execution

**Answer:** b) To ensure that a block of code is always executed, regardless of exceptions

### **Q8: Which of the following is the correct syntax for handling an exception in Python?**

a)

python

Copy code

try:

// code

except Exception:

// handle exception

b)

python

Copy code

try:

code()

catch Exception:

handle\_exception()

c)

python

Copy code

try:

code

else:

exception

d)

python

Copy code

try:

code

finally:

clean up

**Answer:** a)

python

Copy code

try:

// code

except Exception:

// handle exception

### **Q9: What does the raise keyword do in Python?**

a) It raises an exception in the current function  
b) It terminates the program execution  
c) It defines a custom exception  
d) It prints a message

**Answer:** a) It raises an exception in the current function

### **Q10: Which of the following is NOT a built-in exception in Python?**

a) IndexError  
b) ValueError  
c) ConnectionError  
d) IOExceptionError

**Answer:** d) IOExceptionError

### **Q11: How would you catch a specific exception in Python?**

a) Using a generic except block  
b) By specifying the exception type after the except keyword  
c) By using multiple except blocks for the same exception  
d) It is not possible to catch specific exceptions

**Answer:** b) By specifying the exception type after the except keyword

### **Q12: What is the output of the following code?**

python

Copy code

try:

x = 5 / 0

except ZeroDivisionError:

print("Cannot divide by zero")

finally:

print("Execution finished")

a) "Cannot divide by zero"  
b) "Execution finished"  
c) "Cannot divide by zero" "Execution finished"  
d) Error message without any output

**Answer:** c) "Cannot divide by zero" "Execution finished"

### **Q13: Which block ensures that the program is executed without skipping cleanup code, even after exceptions?**

a) finally  
b) except  
c) try  
d) clean

**Answer:** a) finally

### **Q14: What is the correct way to raise a custom exception in Python?**

a) raise Error("Custom error message")  
b) throw Exception("Custom error message")  
c) raise Exception("Custom error message")  
d) raise("Custom error message")

**Answer:** c) raise Exception("Custom error message")

### **Q15: Which of the following is the most general exception class in Python?**

a) BaseException  
b) Exception  
c) SystemExit  
d) Error

**Answer:** b) Exception

### **Q16: What happens if an exception is not caught in Python?**

a) The program will continue execution normally  
b) Python will terminate the program and display an error message  
c) Python will silently ignore the exception  
d) Python will raise an error but continue executing

**Answer:** b) Python will terminate the program and display an error message

### **Q17: What is the purpose of try...except...else structure?**

a) To handle errors only  
b) To run code that may raise an exception and handle it  
c) To run a block of code if no exception occurs  
d) To define a default exception handler

**Answer:** c) To run a block of code if no exception occurs

### **Q18: Which of the following exceptions does the try...except block catch?**

a) Syntax errors  
b) Runtime exceptions  
c) Logical errors  
d) Programmatically defined exceptions only

**Answer:** b) Runtime exceptions

### **Q19: How can you access the message of an exception in Python?**

a) Using error.message()  
b) By using str(exception)  
c) Using getMessage()  
d) Accessing exception.message directly

**Answer:** b) By using str(exception)

### **Q20: What is the syntax to catch multiple exceptions in one except block?**

a)

python

Copy code

except (Error1, Error2):

handle\_error()

b)

python

Copy code

except Error1, Error2:

handle\_error()

c)

python

Copy code

except Error1 or Error2:

handle\_error()

d)

python

Copy code

except [Error1, Error2]:

handle\_error()

**Answer:** a)

python

Copy code

except (Error1, Error2):

handle\_error()

### **Q21: What will happen if the finally block has a return statement?**

a) The finally block is ignored  
b) The return value in the finally block overrides any previous return values  
c) The program will raise an error  
d) It causes an exception

**Answer:** b) The return value in the finally block overrides any previous return values

### **Q22: What is the else block in try...except...else executed for?**

a) When an exception is raised  
b) When an exception is caught  
c) When no exception is raised in the try block  
d) For cleanup code

**Answer:** c) When no exception is raised in the try block

### **Q23: Which of the following exceptions is raised when you try to divide by zero in Python?**

a) ArithmeticError  
b) ZeroDivisionError  
c) ValueError  
d) OverflowError

**Answer:** b) ZeroDivisionError

### **Q24: What is the output of the following code?**

python

Copy code

try:

x = int("abc")

except ValueError:

print("ValueError caught")

finally:

print("Finished execution")

a) "ValueError caught"  
b) "ValueError caught" "Finished execution"  
c) "Finished execution"  
d) "ValueError caught" "ValueError caught"

**Answer:** b) "ValueError caught" "Finished execution"

### **Q25: What type of exception is raised by the following code?**

python

Copy code

try:

f = open("non\_existent\_file.txt")

except FileNotFoundError as e:

print(e)

a) FileNotFoundError  
b) IOError  
c) KeyError  
d) ValueError

**Answer:** a) FileNotFoundError

### **Q26: What is the purpose of try...except block in Python?**

a) To execute code without catching exceptions  
b) To terminate the program when an error occurs  
c) To catch and handle exceptions that may occur during execution  
d) To create custom exceptions

**Answer:** c) To catch and handle exceptions that may occur during execution

### **Q27: What will the following code output?**

python

Copy code

try:

1 / 0

except ZeroDivisionError:

print("Caught ZeroDivisionError")

else:

print("No exception occurred")

a) "Caught ZeroDivisionError"  
b) "No exception occurred"  
c) Error message  
d) No output

**Answer:** a) "Caught ZeroDivisionError"

### **Q28: How do you create a user-defined exception in Python?**

a) By defining a function inside a class  
b) By subclassing the Exception class  
c) By raising a built-in exception  
d) Python does not support user-defined exceptions

**Answer:** b) By subclassing the Exception class

### **Q29: What will the following code output?**

python

Copy code

try:

x = int(input("Enter a number: "))

except ValueError:

print("Invalid input")

finally:

print("End of program")

a) "Invalid input"  
b) "End of program"  
c) Both "Invalid input" and "End of program"  
d) Depends on the user input

**Answer:** d) Depends on the user input

### **Q30: Which exception is raised when an invalid index is accessed in a list in Python?**

a) IndexError  
b) TypeError  
c) ValueError  
d) KeyError

**Answer:** a) IndexError

### **Q31: What is the correct way to catch an exception and display its message?**

a)

python

Copy code

except Exception as e:

print(e)

b)

python

Copy code

except e as Exception:

print(e)

c)

python

Copy code

except Exception:

print(Exception)

d)

python

Copy code

catch Exception as e:

print(e)

**Answer:** a)

python

Copy code

except Exception as e:

print(e)

### **Q32: Which Python keyword is used to raise an exception?**

a) throw  
b) raise  
c) catch  
d) error

**Answer:** b) raise

### **Q33: How do you catch an exception and handle multiple error types separately?**

a) By using multiple except blocks  
b) By using an else block  
c) By using finally block  
d) By using raise and catch

**Answer:** a) By using multiple except blocks

### **Q34: What will happen if there is no except block to handle an exception?**

a) The program will stop with an error message  
b) Python will continue to the next block  
c) Python will ignore the exception  
d) The finally block will still execute

**Answer:** a) The program will stop with an error message

### **Q35: What does the assert statement do in Python?**

a) Raises an exception if the condition is false  
b) Executes code only if a condition is true  
c) Ends the program with a specific message  
d) Ignores errors

**Answer:** a) Raises an exception if the condition is false

### **Q36: What exception is raised when a key is not found in a dictionary?**

a) IndexError  
b) KeyError  
c) TypeError  
d) ValueError

**Answer:** b) KeyError

### **Q37: What will the following code output?**

python

Copy code

try:

x = 1 / 0

except ArithmeticError:

print("Arithmetic error")

finally:

print("Executed")

a) "Arithmetic error"  
b) "Arithmetic error" "Executed"  
c) "Executed"  
d) Error message only

**Answer:** b) "Arithmetic error" "Executed"

### **Q38: Which Python function allows you to catch and display the exception details?**

a) error() b) str()  
c) print()  
d) repr()

**Answer:** b) str()

### **Q39: Which of the following code will correctly raise a custom exception?**

a)

python

Copy code

class CustomError(Exception):

pass

raise CustomError("This is a custom error")

b)

python

Copy code

raise Exception("This is a custom error")

c)

python

Copy code

class CustomError:

pass

raise CustomError("This is a custom error")

d)

python

Copy code

raise CustomError

**Answer:** a)

python

Copy code

class CustomError(Exception):

pass

raise CustomError("This is a custom error")

### **Q40: What type of exception is raised by the code open('file.txt', 'r') if the file does not exist?**

a) FileNotFoundError  
b) IOError  
c) ValueError  
d) FileError

**Answer:** a) FileNotFoundError

### **Q41: Can we use finally block without an accompanying except block?**

a) Yes  
b) No  
c) Only with an else block  
d) Only if there is an error

**Answer:** a) Yes

### **Q42: What is the output of the following code?**

python

Copy code

try:

raise ValueError("This is an error")

except ValueError as ve:

print(ve)

a) This is an error  
b) ValueError  
c) ValueError: This is an error  
d) Error message

**Answer:** c) ValueError: This is an error

### **Q43: Which Python module can be used for exception logging?**

a) os  
b) logging  
c) traceback  
d) sys

**Answer:** b) logging

### **Q44: Which Python method provides a string representation of an exception?**

a) \_\_str\_\_()  
b) exception()  
c) getMessage()  
d) get\_exception()

**Answer:** a) \_\_str\_\_()

### **Q45: What will be printed when the following code executes?**

python

Copy code

try:

x = 10 / 2

except ZeroDivisionError:

print("Error")

else:

print("No error")

a) "Error"  
b) "No error"  
c) No output  
d) Exception traceback

**Answer:** b) "No error"

### **Q46: Which exception is raised by the following code?**

python

Copy code

try:

my\_list = [1, 2, 3]

print(my\_list[5])

except IndexError as e:

print(e)

a) ValueError  
b) IndexError  
c) TypeError  
d) KeyError

**Answer:** b) IndexError

### **Q47: What is the use of the else block in Python's try...except...else?**

a) It is executed if no exception is raised  
b) It is used to raise exceptions  
c) It is executed when an exception is raised  
d) It is used for cleaning up

**Answer:** a) It is executed if no exception is raised

### **Q48: What is the default exception in Python?**

a) RuntimeError  
b) Exception  
c) BaseException  
d) Error

**Answer:** b) Exception

### **Q49: What will the following code output?**

python

Copy code

try:

raise ValueError("Custom exception")

except Exception as e:

print(str(e))

finally:

print("Finally block")

a) Custom exception  
b) Custom exception Finally block  
c) Finally block  
d) Exception traceback

**Answer:** b) Custom exception Finally block

### **Q50: How do you define a custom exception class in Python?**

a) By subclassing the Error class  
b) By subclassing the Exception class  
c) By using the raise keyword  
d) By defining a custom\_exception function

**Answer:** b) By subclassing the Exception class

**Week 10: Python Libraries & Advanced Functionality**

### **Q1: Which of the following Python libraries is used for interacting with the operating system?**

a) sys  
b) os  
c) time  
d) math

**Answer:** b) os

### **Q2: What functionality does the math library provide in Python?**

a) File handling operations  
b) Advanced mathematical functions like trigonometric and logarithmic functions  
c) String manipulation  
d) Time-related operations

**Answer:** b) Advanced mathematical functions like trigonometric and logarithmic functions

### **Q3: Which Python library can be used to work with dates and times?**

a) math  
b) time  
c) random  
d) os

**Answer:** b) time

### **Q4: What is the primary purpose of the sys library in Python?**

a) To handle dates and times  
b) To provide access to system-specific parameters and functions  
c) To perform advanced mathematical operations  
d) To interact with the operating system

**Answer:** b) To provide access to system-specific parameters and functions

### **Q5: Which of the following is NOT a feature of the os library in Python?**

a) File manipulation  
b) Directory management  
c) Network communication  
d) Environment variable management

**Answer:** c) Network communication

### **Q6: In Python, which library is commonly used for machine learning and data analysis?**

a) math  
b) os  
c) scikit-learn  
d) sys

**Answer:** c) scikit-learn

### **Q7: What is the primary function of socket programming in Python?**

a) Handling files  
b) Creating network connections and communication between clients and servers  
c) Handling mathematical operations  
d) Handling dates and times

**Answer:** b) Creating network connections and communication between clients and servers

### **Q8: What is a typical use case of the socket library in Python?**

a) File I/O operations  
b) Interacting with databases  
c) Building network-based applications like web servers or chat applications  
d) Sorting data in lists

**Answer:** c) Building network-based applications like web servers or chat applications

### **Q9: What does "client-server architecture" refer to?**

a) A type of database structure  
b) A programming design pattern where clients request services from a central server  
c) A method for debugging code  
d) A function for managing system processes

**Answer:** b) A programming design pattern where clients request services from a central server

### **Q10: Which Python function is used to get the current time?**

a) os.time()  
b) sys.time()  
c) time.time()  
d) time.current\_time()

**Answer:** c) time.time()

### **Q11: How do Python plugins work?**

a) They are executable files that run independently  
b) They allow modular and extendable code by loading additional functionality at runtime  
c) They help in optimizing the Python interpreter  
d) They are used to analyze exploit code

**Answer:** b) They allow modular and extendable code by loading additional functionality at runtime

### **Q12: In Python, what is typically used to create modular code with plugins?**

a) Classes  
b) Functions  
c) Modules and packages  
d) Loops

**Answer:** c) Modules and packages

### **Q13: What is an example of an exploit analysis automation task that can be done using Python?**

a) Automatically analyzing vulnerabilities in code  
b) Creating network protocols  
c) Sorting and filtering data  
d) Compressing files

**Answer:** a) Automatically analyzing vulnerabilities in code

### **Q14: Which debugging tool in Python is used for interactive debugging and stepping through code?**

a) logging  
b) pdb  
c) sys  
d) math

**Answer:** b) pdb

### **Q15: Which Python module helps in logging errors and messages for debugging purposes?**

a) sys  
b) logging  
c) socket  
d) math

**Answer:** b) logging

### **Q16: Which of the following libraries is widely used for numerical computations and data manipulation in Python?**

a) time  
b) numpy  
c) os  
d) socket

**Answer:** b) numpy

### **Q17: Which library is commonly used for data manipulation and analysis, especially for tabular data?**

a) matplotlib  
b) numpy  
c) pandas  
d) time

**Answer:** c) pandas

### **Q18: Which Python library is used for creating visualizations like graphs and charts?**

a) numpy  
b) matplotlib  
c) pandas  
d) time

**Answer:** b) matplotlib

### **Q19: Which library is the foundation for machine learning in Python, providing tools for regression, classification, and clustering?**

a) scikit-learn  
b) pandas  
c) numpy  
d) matplotlib

**Answer:** a) scikit-learn

### **Q20: Which Python function is used to create a new socket connection for network communication?**

a) socket.create()  
b) socket.open()  
c) socket.socket()  
d) socket.connect()

**Answer:** c) socket.socket()

### **Q21: Which of the following is NOT an advantage of using Python for exploit analysis automation?**

a) Scripting and automating repetitive tasks  
b) Fast execution  
c) Easy integration with other tools and libraries  
d) Flexibility in writing custom scripts for analysis

**Answer:** b) Fast execution

### **Q22: How can Python help in debugging code?**

a) By providing a debugger (pdb) and logging functionality  
b) By ignoring errors and continuing execution  
c) By automatically fixing bugs in the code  
d) By optimizing the code execution speed

**Answer:** a) By providing a debugger (pdb) and logging functionality

### **Q23: Which Python library can help in handling HTTP requests and responses when building client-server applications?**

a) socket  
b) os  
c) http  
d) time

**Answer:** c) http

### **Q24: What is the purpose of the pdb library in Python?**

a) To handle input/output operations  
b) To debug code interactively by setting breakpoints and stepping through code  
c) To manipulate time and dates  
d) To perform statistical analysis

**Answer:** b) To debug code interactively by setting breakpoints and stepping through code

### **Q25: Which of the following Python libraries would you use for matrix and vector operations?**

a) math  
b) numpy  
c) sys  
d) logging

**Answer:** b) numpy

### **Q26: What does the time library in Python provide?**

a) Mathematical functions  
b) Functions for manipulating date and time  
c) Socket programming  
d) Logging features

**Answer:** b) Functions for manipulating date and time

### **Q27: In client-server architecture, what role does the server play?**

a) It sends requests to clients  
b) It receives and processes requests from clients  
c) It handles data storage  
d) It acts as a client

**Answer:** b) It receives and processes requests from clients

### **Q28: What type of applications can be built using Python's socket programming?**

a) Simple text-based applications  
b) Networked applications like web servers, chat applications, etc.  
c) Database management systems  
d) Data analysis tools

**Answer:** b) Networked applications like web servers, chat applications, etc.

### **Q29: What is the logging module used for in Python?**

a) To log function calls  
b) To track and record events, errors, or debugging information during program execution  
c) To handle mathematical calculations  
d) To create network connections

**Answer:** b) To track and record events, errors, or debugging information during program execution

### **Q30: How can you run a machine learning model using Python?**

a) By using the numpy library  
b) By using scikit-learn for training and prediction  
c) By using the os library  
d) By manually writing complex algorithms

**Answer:** b) By using scikit-learn for training and prediction

### **Q31: What kind of information can the sys library provide?**

a) Mathematical functions  
b) Time-related functions  
c) System-specific parameters and functions like command-line arguments  
d) File manipulation operations

**Answer:** c) System-specific parameters and functions like command-line arguments

### **Q32: What is the primary purpose of the scikit-learn library?**

a) To create network connections  
b) To build machine learning models for tasks like classification and regression  
c) To handle mathematical operations  
d) To visualize data

**Answer:** b) To build machine learning models for tasks like classification and regression

### **Q33: What is the function of the socket.connect() method?**

a) To start the server  
b) To create a new socket  
c) To connect a client socket to a server  
d) To send data to a socket

**Answer:** c) To connect a client socket to a server

### **Q34: What is the key advantage of Python's logging library?**

a) It allows real-time code execution without errors  
b) It stores log messages to track program behavior  
c) It speeds up the code execution  
d) It automatically corrects code errors

**Answer:** b) It stores log messages to track program behavior

### **Q35: Which library would you use for handling matrix operations and linear algebra tasks in Python?**

a) os  
b) math  
c) numpy  
d) time

**Answer:** c) numpy

### **Q36: How can Python be used to automate the exploit analysis process?**

a) By creating manual scripts  
b) By using machine learning for classification  
c) By using Python to develop scripts that can analyze code vulnerabilities automatically  
d) By using Python to run brute force attacks

**Answer:** c) By using Python to develop scripts that can analyze code vulnerabilities automatically

### **Q37: Which Python tool can be used to debug and trace Python code step by step?**

a) pdb  
b) logging  
c) math  
d) sys

**Answer:** a) pdb

### **Q38: Which library is NOT typically used for machine learning in Python?**

a) scikit-learn  
b) tensorflow  
c) numpy  
d) logging

**Answer:** d) logging

### **Q39: What is the role of pandas in data analysis?**

a) It provides mathematical functions  
b) It handles data visualization  
c) It provides data structures like DataFrame for handling structured data  
d) It is used for web scraping

**Answer:** c) It provides data structures like DataFrame for handling structured data

### **Q40: Which of the following is true about client-server architecture in Python?**

a) The client is always the server  
b) The server sends requests to the client  
c) The server listens for client requests and sends back responses  
d) Both client and server are identical

**Answer:** c) The server listens for client requests and sends back responses

### **Q41: What can be used to add real-time logging features to a Python application?**

a) socket  
b) logging  
c) sys  
d) math

**Answer:** b) logging

### **Q42: What is the typical role of plugins in Python?**

a) To speed up the execution time of code  
b) To add optional, reusable features to the program without modifying the core functionality  
c) To handle exceptions  
d) To define global variables

**Answer:** b) To add optional, reusable features to the program without modifying the core functionality

### **Q43: Which Python library is used to handle simple operations like rounding, factorial, and power?**

a) math  
b) os  
c) sys  
d) time

**Answer:** a) math

### **Q44: What is the purpose of the raise keyword in Python?**

a) To pause execution  
b) To raise a custom exception  
c) To define a new variable  
d) To log an error

**Answer:** b) To raise a custom exception

### **Q45: What is a key use of Python in debugging?**

a) Ignoring errors  
b) Using logging for error tracking and pdb for stepping through code  
c) Automatically fixing errors  
d) Writing complex algorithms for error detection

**Answer:** b) Using logging for error tracking and pdb for stepping through code

### **Q46: What is the advantage of using Python for network programming?**

a) It is faster than other languages  
b) It has built-in libraries for network communication like socket  
c) It requires no external libraries  
d) It automatically handles security vulnerabilities

**Answer:** b) It has built-in libraries for network communication like socket

### **Q47: What will happen if you import the os library in Python?**

a) You can interact with files, directories, and system variables  
b) It helps with network programming  
c) It provides graphical user interface functionality  
d) It handles time-related tasks

**Answer:** a) You can interact with files, directories, and system variables

### **Q48: How can Python assist in exploit analysis automation?**

a) By analyzing network traffic  
b) By performing system security checks  
c) By automating the analysis of vulnerabilities using predefined scripts  
d) By creating secure encryption algorithms

**Answer:** c) By automating the analysis of vulnerabilities using predefined scripts

### **Q49: How do you define a new socket in Python?**

a) socket.create\_socket()  
b) socket.connect()  
c) socket.socket()  
d) socket.define()

**Answer:** c) socket.socket()

### 

### **Q50: What is an important feature of the scikit-learn library?**

a) Web scraping  
b) Data manipulation  
c) Machine learning for tasks like classification, regression, and clustering  
d) Visualization

**Answer:** c) Machine learning for tasks like classification, regression, and clustering