**Python Data Structures and Functions Test**

**Total Questions:** 30 (20 MCQs, 10 Coding Challenges, Total 80 MARKS)

**Time Limit**: 60 minutes

**Section 1: Multiple Choice Questions (20 Questions, 2 Marks Each)**

**Strings (2 Questions)**

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

s = “PythonProgramming”

print(s[2:10:2])

a) toPorm

b) tPg

c) toPg

d) hnrg

ANSWER: A

2. What does “Hello”.replace(‘1’, ‘ ‘)return?

a) Heo

b) Helo

c) Hello

d) H

ANSWER: C

**Lists and Tuples (3 Questions)**

3. What will be the output?

lst = [1, 2, 3]

lst.extend([4, 5])

print(lst)

a) [1, 2, 3, [4, 5]]

b) [1, 2, 3, 4, 5]

c) [1, 2, 3]

d) [1, 2, 3, (4, 5)]

**ANSWER**: B

4. Which of the following statements is incorrect regarding tuples?

a) Tuples are immutable.

b) Tuples consume less memory than lists.

c) Tuples support item assignment.

d) Tuples can contain mutable objects.

**ANSWER**: C

5. What will tuple([1, 2, 3, 4]) return?

a) (1, 2, 3, 4)

b) [1, 2, 3, 4]

c) {1, 2, 3, 4}

d) None

**ANSWER**: A

**List Slicing (2 Questions)**

6. What does lst[-3:-1] return for lst = [10, 20, 30, 40, 50] ?

a) [30, 40]

b) [40, 50]

c) [30, 40, 50]

d) [20, 30]

**ANSWER**: A

7. What does list[::-1] return for list =[‘a’, ‘b’, ‘c’, ‘d’] ?

a) [‘a’, ‘b’, ‘c’, ‘d’]

b) [‘d, ‘c, ‘b, ‘a’]

c) [‘c’, ‘d’]

d) [‘b’, ‘a’]

**ANSWER**: B

**List Comprehension (2 Questions)**

8. What does [x\*\*2 for x in range(5) if x % 2 == 0] return?

a) [0, 4, 16]

b) [1, 4, 9, 16]

c) [0, 1, 4, 9, 16]

d) [0, 2, 4]

**ANSWER**: A

9. Identify the incorrect list comprehension syntax:

a) [x for x in range(10) if x % 2 == 0]

b)[x for x in range(10) x % 2 == 0]

c) [x+1 for x in range(3)]

d) [x for x in range(3) if x &gt; 1]

**ANSWER**: A

**Range (2 Questions)**

10. What is the output of range(1, 5, 2) in list form?

a) [1, 3]

b) [1, 2, 3, 4]

c) [1, 2, 3, 4, 5]

d) [1, 2, 4]

**ANSWER**: A

11. What happens if range(5, 1, 1) is executed?

a) [5, 4, 3, 2, 1]

b) [5, 4, 3, 2]

c) []

d) [5]

**ANSWER**: C

**Dictionaries & Dictionary Comprehension (3 Questions)**

12. What is the output of the below?

d ={‘x’:1, ‘y’: 2}

d.update({‘y’:3, ‘z’:4})

print(d)

a) {‘x’:1, ‘y’:3, ‘z’:4}

b) {‘x’:1, ‘y’:2, ‘z’:4}

c) {‘x’:1, ‘y’:2}

d) {‘x’:1, ‘z’:4}

**ANSWER**: A

13. What does {x: x\*\*3 for x in range(2, 5)} return?

a) {2: 8, 3: 27, 4: 64}

b) {2: 4, 3: 9, 4: 16}

c) {2: 6, 3: 9, 4: 12}

d) {2: 3, 3: 6, 4: 9}

**ANSWER**: A

14. What will list(d.keys())[0] return for d = {‘a’:100, ‘b’:200} ?

a) 100

b) ‘a’

c) ’b’

d) None

**ANSWER**: A

**Functions (6 Questions)**

15. What is the output?

def f(x, y=[]):

y.append(x)

return y

print(f(1))

print(f(2))

a) [1] [2]

b) [1] [1, 2]

c) [1] []

d) Error

**ANSWER**: B

16. What will lambda x: x \* x return when x = 3 ?

a) 6

b) 9

c) 3

d) None

**ANSWER**: B

17. Which function is used to filter elements from an iterable?

a) map()

b) reduce()

c) filter()

d) apply()

**ANSWER**: C

18. What is the output?

from functools import reduce

reduce(lambda x, y: x \* y, [1, 2, 3, 4])

a) 10

b) 24

c) [1, 2, 3, 4]

d) None

**ANSWER:** C

19. What does filter(lambda x: x >2, [1, 2, 3, 4]) return?

a) [3, 4]

b) [1, 2]

c) [1, 2, 3, 4]

d) []

**ANSWER**: A

20. Write a function that accepts a variable number of arguments and prints them.

**ANSWER**:

def accept\_variable(\*args):

for arg in args:

print(arg)

accept\_variable(‘hello’, 1, .9, ‘world’)

**Section 2: Coding Challenges (10 Questions, 4 Marks Each)**

1. Reverse a string without using slicing

def reverse\_string(s):

# Your code here

# Example

print(reverse\_string(“Python”))

Expected Output: “nohtyP”

**ANSWER**:

def reverse\_string(s):

reversed\_str = ""

for char in s:

reversed\_str = char + reversed\_str

return reversed\_str

print(reverse\_string("Python"))

2. Write a function to remove duplicates from a list

def remove\_duplicates(lst):

# Your code here

# Example

print(remove\_duplicates([1, 2, 2, 3, 4, 4, 5]))

Expected Output:[1, 2, 3, 4, 5]

**ANSWER**:

def remove\_duplicates(lst):

unique\_list = []

for item in lst:

if item not in unique\_list:

unique\_list.append(item)

return unique\_list

print(remove\_duplicates([1, 2, 2, 3, 4, 4, 5]))

3. Write a dictionary comprehension that reverses keys and values

def reverse\_dict(d):

# Your code here

# Example

print(reverse\_dict({‘a’ : 1, ‘b’ : 2, ‘c’ : 3}))

Expected Output:{1: ‘a’, 2: ‘b’, 3:’c’}

**ANSWER**:

def reverse\_dict(d):