**Python Function exercises solution**

1. **Rectangle Area**

def rectangle\_area(width=10, height=5):

return width \* height

print(rectangle\_area()) # 50

1. **Repeat String**

def repeat\_string(s, n=1):

return s \* n

1. **Compound Interest**

def compound\_interest(principal, rate=5, time=10):

return principal \* (1 + rate / 100) \*\* time

1. **Nth Fibonacci Number**

def fibonacci(n=10):

a, b = 0, 1

for \_ in range(n):

a, b = b, a + b

return a

1. **Celsius to Fahrenheit**

def celsius\_to\_fahrenheit(celsius=0):

return (celsius \* 9/5) + 32

1. **Volume of Cylinder**

def cylinder\_volume(radius=5, height=10):

import math

return math.pi \* radius \*\* 2 \* height

1. **Check Even Number**

def is\_even(number=2):

return number % 2 == 0

1. **Current Date and Time**

def current\_datetime(format\_str="%Y-%m-%d %H:%M:%S"):

from datetime import datetime

return datetime.now().strftime(format\_str)

1. **List of Squares**

def list\_of\_squares(n=10):

return [i\*\*2 for i in range(1, n+1)]

1. **Random Numbers List**

def random\_numbers\_list(size=10, start=1, end=100):

import random

return [random.randint(start, end) for \_ in range(size)]

1. **Total Cost of Order**

def total\_cost(price, quantity, discount=0):

return (price \* quantity) \* ((100 - discount) / 100)

1. **Student Information**

def student\_info(name, age, grade):

return f"Name: {name}, Age: {age}, Grade: {grade}"

1. **Rectangle Perimeter**

def rectangle\_perimeter(length, width):

return 2 \* (length + width)

1. **String Formatter**

def format\_string(s, prefix="", suffix=""):

return f"{prefix}{s}{suffix}"

1. **Maximum of Three Numbers**

def max\_of\_three(a, b, c):

return max(a, b, c)

1. **Formatted Address**

def address(street, city, zip\_code):

return f"{street}, {city}, {zip\_code}"

1. **Calculate BMI**

def calculate\_bmi(weight, height):

return weight / (height \*\* 2)

1. **Distance Between Points**

def distance(x1, y1, x2, y2):

return ((x2 - x1)\*\*2 + (y2 - y1)\*\*2)\*\*0.5

1. **User Profile**

def user\_profile(username, email, password):

return {"username": username, "email": email, "password": password}

1. **Greeting Message**

def greeting(first\_name, last\_name):

return f"Hello, {first\_name} {last\_name}!"

**Positional Arguments**

1. **GCD of Two Numbers**

def gcd(a, b):

while b:

a, b = b, a % b

return a

1. **Element at Given Position**

def get\_element(lst, pos):

return lst[pos]

1. **Average of List**

def average(lst):

return sum(lst) / len(lst)

1. **Dot Product of Vectors**

def dot\_product(a, b):

return sum(x \* y for x, y in zip(a, b))

1. **Swap Values**

def swap(a, b):

return b, a

1. **Distance Traveled**

def distance\_traveled(speed, time):

return speed \* time

1. **Median of Three Numbers**

def median(a, b, c):

return sorted([a, b, c])[1]

1. **Intersection of Sets**

def intersection(set1, set2):

return set1 & set2

1. **Multiply List Elements**

def multiply\_list(lst, num):

return [x \* num for x in lst]

1. **Sum of Positional Arguments**

def sum\_all(\*args):

return sum(args)

1. **Print Keyword Arguments**

def print\_kwargs(\*\*kwargs):

for key, value in kwargs.items():

print(f"{key}: {value}")

1. **Maximum Value using args**

def max\_value(\*args):

return max(args)

1. **Concatenate Strings using args**

def concatenate\_all(\*args):

return "".join(args)

1. **Merge Dictionaries using kwargs**

def merge\_dicts(\*\*kwargs):

result = {}

for d in kwargs.values():

result.update(d)

return result

1. **Print args and kwargs**

def print\_all(\*args, \*\*kwargs):

print("Positional arguments:", args)

print("Keyword arguments:", kwargs)

print\_all(1, 2, 3, a=4, b=5)

1. **Product of Positional Arguments**

def product\_all(\*args):

result = 1

for num in args:

result \*= num

return result

1. **Sum of Keyword Argument Values**

def sum\_kwargs\_values(\*\*kwargs):

return sum(kwargs.values())

1. **Longest String using args**

def longest\_string(\*args):

return max(args, key=len)

1. **Concatenate Lists using args**

def concatenate\_lists(\*args):

result = []

for lst in args:

result.extend(lst)

return result

1. **Tuple of Squares using args**

def squares\_tuple(\*args):

return tuple(x\*\*2 for x in args)

1. **Filter Even Numbers using args**

def filter\_evens(\*args):

return [x for x in args if x % 2 == 0]

1. Uppercase Keys using **kwargs**

def uppercase\_keys(\*\*kwargs):

return {k.upper(): v for k, v in kwargs.items()}

1. **Average of Positional Arguments**

def average\_all(\*args):

return sum(args) / len(args) if args else 0

1. **Keys and Values using kwargs**

def keys\_and\_values(\*\*kwargs):

return list(kwargs.items())

1. **Flatten Lists using args**

python

Copy code

def flatten\_lists(\*args):

return [item for sublist in args for item in sublist]

1. Concatenate Values using **kwargs**

def concatenate\_values(\*\*kwargs):

return ''.join(map(str, kwargs.values()))

48. Union of Sets using \*args\*\*

def union\_sets(\*args):

result = set()

for s in args:

result |= s

return result

49. \*\*Reverse Print of Arguments\*\*

def reverse\_print(\*args, \*\*kwargs):

print("Positional arguments (reversed):", \*reversed(args))

print("Keyword arguments (reversed):", \*\*{k: v for k, v in kwargs.items()})

**50. Sort Keys in Dictionary**

def sort\_keys(\*\*kwargs):

return {k: kwargs[k] for k in sorted(kwargs)}