

# **ANSWER KEY**

## **PYTHON WORKBOOK – SECTION 5**

***Functions & Modular Programming***

Programmer's Hub – by CodeWithVivek  
<https://www.youtube.com/@code-with-vivek>

## 5.1 What Are Functions?

### Try This – Sample Solution

Why are functions important?

1. They make code reusable
  2. They make code clean and organized
  3. They help avoid repetition
  4. They make debugging easier
- 

## 5.2 Defining & Calling Functions

### Try This – Sample Solution

```
def welcome():  
    print("Welcome to Python Programming!")
```

### Debug This – Answer

```
def say_hello  
    print("Hello!")
```

Missing colon (:) after function name

Proper version:

```
def say_hello():  
    print("Hello!")
```

---

## 5.3 Parameters & Return Values

### Your Turn – Sample Solution

```
def add_numbers(a, b):  
    print("Sum =", a + b)
```

### Concept Check

- **Parameter:** The variable listed in the function definition
- **Argument:** The value passed when calling the function

Example:

```
def greet(name): # name = parameter  
    print(name)
```

```
greet("Vivek") # "Vivek" = argument
```

---

### Return Values

#### Try This – Sample Solution

```
def square(n):  
    return n * n
```

#### Debug This – Explanation

```
def multiply(a, b):  
    print(a * b)
```

```
x = multiply(4, 5)
```

```
print(x + 10)
```

`multiply()` prints the result but does **not return** anything

So `x` becomes **None**, causing error when doing `x + 10`

Correct version:

```
def multiply(a, b):  
    return a * b
```

## Default Parameters

### Your Turn – Sample Solution

```
def power(base, exponent=2):  
    return base ** exponent
```

---

## Keyword & Positional Arguments

### Try This – Sample Solutions

#### Positional:

```
intro("Vivek", "Delhi")
```

#### Keyword:

```
intro(city="Mumbai", name="Amit")
```

---

## Variable-Length Arguments

### \*args – Sample Solution

```
def largest(*nums):
```

```
    print(max(nums))
```

### \*\*kwargs – Sample Solution

```
def student_record(**info):
```

```
    for key, value in info.items():
```

```
        print(key, ":", value)
```

```
student_record(name="Vivek", age=25, city="Delhi")
```

---

## 5.4 Lambda Functions

### Try This – Sample Solution

```
last_char = lambda s: s[-1]  
  
print(last_char("Python"))
```

## 5.5 Map, Reduce, Filter

### map(): Your Turn – Sample Code

```
numbers = list(range(1, 11))

result = list(map(lambda x: x * 10, numbers))

print(result)
```

### filter(): Your Turn – Sample Code

```
numbers = [10, 45, 60, 23, 80, 55, 30, 100]

filtered_numbers = list(filter(lambda x: x > 50, numbers))

print(filtered_numbers)
```

### reduce(): Your Turn – Sample Code

```
from functools import reduce

numbers = [2, 3, 4, 5]

product = reduce(lambda a, b: a * b, numbers)

print(product)
```

---

## 5.6 Modules & Packages

### Your Turn – Sample Code

```
import datetime
```

```
today = datetime.date.today()

print(today)
```

## 5.9 Practice Problems – Instructor Solutions

### Problem 1 – Even or Odd

```
def even_odd(n):  
    if n % 2 == 0:  
        return "Even"  
    return "Odd"
```

---

### Problem 2 – Factorial

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
    return result
```

---

### Problem 3 – Filter Even Numbers

```
def get_evens(lst):  
    return [x for x in lst if x % 2 == 0]
```

---

### Problem 4 – Count Vowels

```
def count_vowels(s):  
    vowels = "aeiouAEIOU"  
    count = 0  
    for ch in s:  
        if ch in vowels:  
            count += 1  
    return count
```

**Mini Assignment – Full Instructor Solution****math\_utils.py**

```
def is_prime(n):  
    if n <= 1:  
        return False  
  
    for i in range(2, int(n**0.5) + 1):  
        if n % i == 0:  
            return False  
  
    return True  
  
  
def gcd(a, b):  
    while b != 0:  
        a, b = b, a % b  
  
    return a  
  
  
def lcm(a, b):  
    return a * b // gcd(a, b)
```

---

**main.py**

```
import math_utils  
  
  
print("Prime:", math_utils.is_prime(17))  
print("GCD:", math_utils.gcd(12, 18))  
print("LCM:", math_utils.lcm(12, 18))
```