

<https://github.com/ssardarabdullah>

ABBOTTABAD UNIVERSITY OF SCIENCE & TECHNOLOGY
ABBOTTABAD



Lab Task No # 02

NAME :

Abdullah Niaz

SUBMITTED TO:

SIR JAMAL ABDUL AHAD

**COURSE:
PROGRAM**

**OBJECT ORIENTED
BS/SC**

SECTION :

C

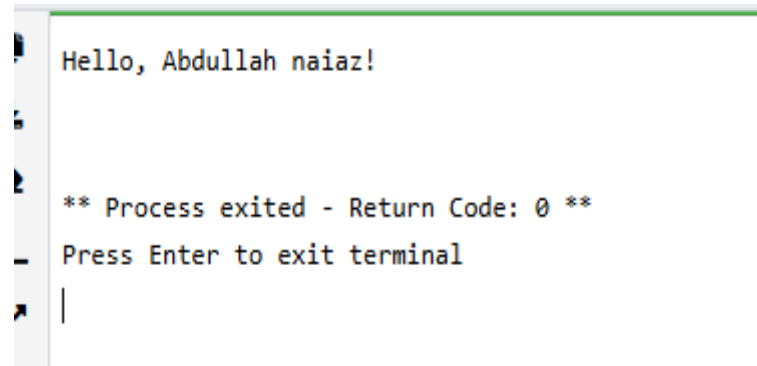
ROLL NO :

F24- 1312

i-:

```
def hello_name(name):  
    print(f"Hello, {name}!")  
  
hello_name("Abdullah naiaz")
```

Output:

A terminal window with a light gray background and a green title bar. The output of the Python code is displayed in a monospaced font. The first line is "Hello, Abdullah naiaz!". The second line is "** Process exited - Return Code: 0 **". The third line is "Press Enter to exit terminal". The fourth line shows a vertical cursor bar.

```
Hello, Abdullah naiaz!  
  
** Process exited - Return Code: 0 **  
Press Enter to exit terminal  
|
```

ii-:

```
shanaab.py - C:/Users/MIH/AppData/Local/Programs/Python/Python313/shanaab.py (3.13.2)
File Edit Format Run Options Window Help
def calculate_area(length, width=10):
    return length * width
|
areal = calculate_area(5)
area2 = calculate_area(5, 20)

print(areal)
print(area2)
```

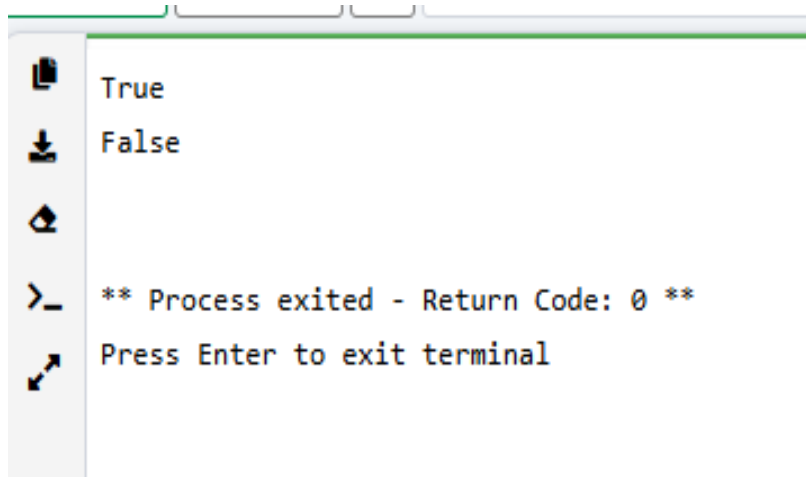
Output:

```
Run Share Command Line Arguments
50
100
** Process exited - Return Code: 0 **
Press Enter to exit terminal
|
```

iii-:

```
def is_even(number):  
    return number % 2 == 0  
  
print(is_even(4))  
print(is_even(7))  
|
```

OUTPUT:



```
True  
False  
  
** Process exited - Return Code: 0 **  
Press Enter to exit terminal
```

iv-:

```
File Edit Format Run Options Window Help
count = 0

def increment_count():
    global count
    count += 1
    print("Count inside function:", count)
increment_count()
print("Count outside function:", count)
```

OUTPUT:

```
Run Share Command Line Arguments
Count inside function: 1
Count outside function: 1
Start Terminal exited - Return Code: 0 **
Press Enter to exit terminal
```

V-:

Input :

```
File Edit Format Run Options Window Help
def fibonacci(n):
    if n <= 0:
        return "Input must be a positive integer"
    elif n == 1:
        return 0
    elif n == 2:
        return 1
    return fibonacci(n - 1) + fibonacci(n - 2)

print(fibonacci(10)) # Output: 34
```

OUTPUT:

```
run Share Command Line Arguments
34
** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

vi-:

```
File Edit Format Run Options Window Help
is_even = lambda x: x % 2 == 0
|
print(is_even(4))
print(is_even(7))
```

OUTPUT:

```
True
False

** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

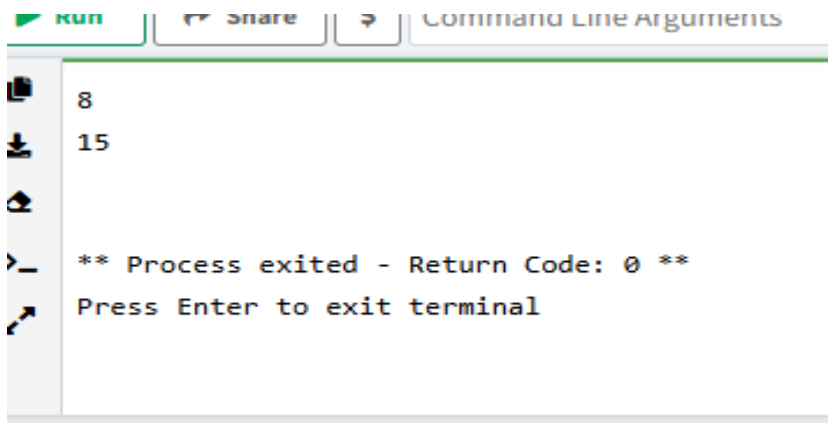
vii-:

INPUT:

File Edit Format Run Options Window Help

```
def operate(func, a, b):  
    return func(a, b)  
add = lambda x, y: x + y  
multiply = lambda x, y: x * y  
  
print(operate(add, 5, 3))  
print(operate(multiply, 5, 3))  
|
```

OUTPUT



```
Run Share Command Line Arguments  
8  
15  
  
** Process exited - Return Code: 0 **  
Press Enter to exit terminal
```

viii:

their product

INPUT:

```
File Edit Format Run Options Window Help
def multiply_all(*args):
    result = 1
    for num in args:
        result *= num
    return result
print(multiply_all(2, 3, 4))
print(multiply_all(5, 10))
print(multiply_all(7))
print(multiply_all())
```

OUTPUT:

```
24
50
7
1
>_
↗
```

ix-:

INPUT:

```
def log_decorator(func):  
    def wrapper(*args, **kwargs):  
        print("Function called")  
        return func(*args, **kwargs)  
    return wrapper  
@log_decorator  
def say_hello():
```

OUTPUT:

```
Function called  
Hello!  
  
** Process exited - Return Code: 0 **  
Press Enter to exit terminal
```