

OBJECT ORIENTED PROGRAMMING



LAB TASK 2

NAME: Hassan Khan

ROLL NO: F24_605

SEMESTER: 2nd

SECTION: C

SUBJECT: Object oriented programming

SUBMITTED TO: Mr. Jamal Abdul Ahad

DEPARTMENT OF COMPUTER SCIENCE

**ABBOTTABAD UNIVERSITY OF SCIENCE AND
TECHNOLOGY**

Functions in Python

Part 1: Introduction to Functions

What is a Function?

A function is a block of reusable code that performs a specific task. Functions help in organizing code and improving reusability.

Defining a Function

A function is defined using the `def` keyword. # Example `def greet(): print("Hello, World!")` `greet()` # Calling the function

Exercise 1:

Write a function `hello_name` that takes a name as an argument and prints "Hello, !".



```
File Edit Format Run Options Window Help
def hello_name(name):
    print(f"Hello {name}!")
hello_name("hassan") # Output: Hello hassan!
```

```
IDLE Shell 3.13.2
File Edit Shell Debug Options Window Help
Python 3.13.2 (tags/v3.13.2:4ff8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python313/hello word.py ==
Hello hassan!
>>> |
```

Part 2: Function Arguments

Positional and Keyword Arguments

Functions can have parameters that receive values when called.

Positional Arguments

```
def add(a, b):  
  
    return a + b  
  
# Keyword Arguments  
  
print(add(a=3, b=5))
```

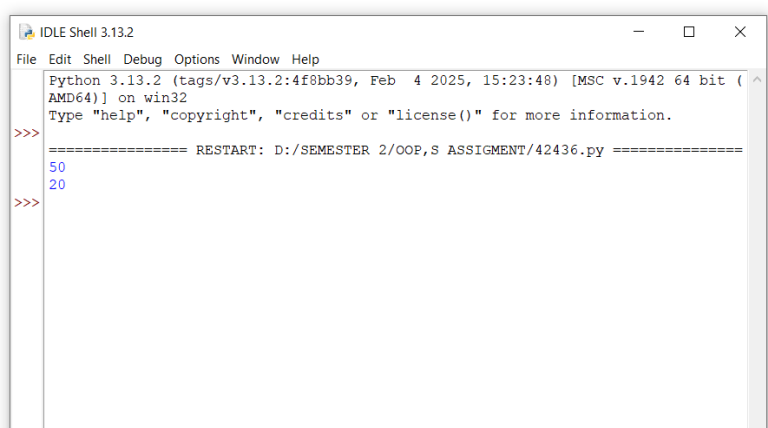
Default Parameters

```
# Default value for a parameter  
  
def power(base, exp=2):  
  
    return base ** exp  
  
print(power(3)) # Uses default exp=2  
  
print(power(3, 3))
```

Exercise 2:

Write a function `calculate_area` that takes `length` and `width` as arguments and returns the area of a rectangle. The `width` should have a default value of 10.

```
*42436.py - D:/SEMESTER 2/OOP,S ASSIGNMENT/42436.py (3.13.2)*  
File Edit Format Run Options Window Help  
def calculate_area(length, width=10):  
    return length * width  
print(calculate_area(5))  
print(calculate_area(5, 4))
```



```
IDLE Shell 3.13.2  
File Edit Shell Debug Options Window Help  
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb 4 2025, 15:23:40) [MSC v.1942 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: D:/SEMESTER 2/OOP,S ASSIGNMENT/42436.py =====  
50  
20  
>>>
```

Part 3: Return Values

A function can return a value using the return keyword.

```
def square(num):
```

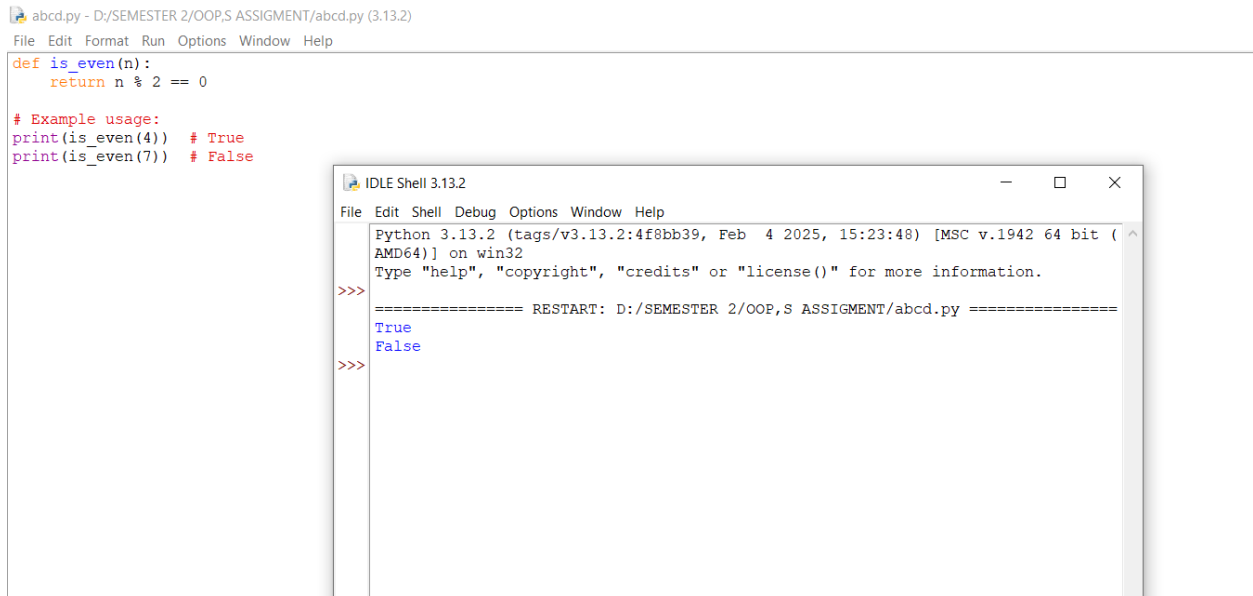
```
    return num * num
```

```
result = square(4)
```

```
print(result) # 16
```

Exercise 3:

Write a function `is_even` that returns `True` if a given number is even and `False` otherwise.



The screenshot shows the Python IDLE 3.13.2 environment. The main editor window displays the following code:

```
abcd.py - D:/SEMESTER 2/OOP,S ASSIGNMENT/abcd.py (3.13.2)
File Edit Format Run Options Window Help
def is_even(n):
    return n % 2 == 0

# Example usage:
print(is_even(4)) # True
print(is_even(7)) # False
```

The Shell window shows the execution results:

```
IDLE Shell 3.13.2
File Edit Shell Debug Options Window Help
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/SEMESTER 2/OOP,S ASSIGNMENT/abcd.py =====
True
False
>>>
```

Part 4: Variable Scope

Local and Global Variables

Variables defined inside a function are local, while those outside are global.

```
global_var = "I am global"
```

```
def my_function():
```

```
    local_var = "I am local"
```

```
print(global_var)
```

```
print(local_var)
```

```
my_function()
```

```
print(global_var)
```

```
# print(local_var) # This will cause an error
```

Exercise 4:

Create a function that modifies a global variable inside a function using the global keyword.

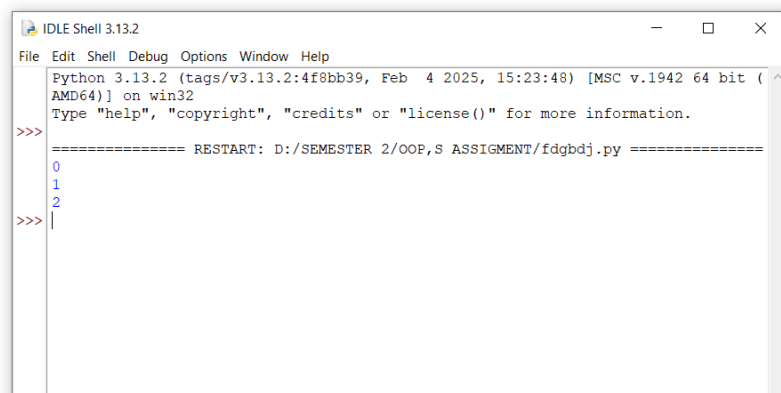
fdgbdj.py - D:/SEMESTER 2/OOP,S ASSIGNMENT/fdgbdj.py (3.13.2)

File Edit Format Run Options Window Help

```
# Define a global variable
counter = 0

def increase_counter():
    global counter # Declare that we are using the global variable
    counter += 1 # Modify the global variable

# Example usage
print(counter) # Output: 0
increase_counter()
print(counter) # Output: 1
increase_counter()
print(counter) # Output: 2
```



```
IDLE Shell 3.13.2
File Edit Shell Debug Options Window Help
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/SEMESTER 2/OOP,S ASSIGNMENT/fdgbdj.py =====
0
1
2
>>> |
```