Experiment 7

**Aim** - Write python programs to implement Functions (Built-in, User defined, Anonymous)

**Theory:**

**1. Built-In Functions:**

Python comes with a rich set of built-in functions that provide fundamental operations and

functionalities. Here's an overview of some of the commonly used built-in functions in Python:

● len(iterable): Returns the length (number of items) of an iterable object, such as a string, list,

tuple, etc.

● type(object): Returns the type of the object.

● print(\*objects, sep=' ', end='\n', file=sys.stdout, flush=False): Prints the specified objects to

the standard output (usually the console).

● input(prompt=''): Reads a line from the standard input (usually the console) and returns it as a

string.

● int(x, base=10), float(x), str(object): Convert a number or string to an integer, float, or string,

respectively.

● list(iterable), tuple(iterable), set(iterable): Creates a list, tuple, or set from the elements of an

iterable.

● max(iterable, \*args, key=None), min(iterable, \*args, key=None): Returns the maximum or

minimum element from an iterable (or among the arguments).

● sum(iterable, start=0): Returns the sum of all elements in an iterable, with an optional start

value.

● sorted(iterable, key=None, reverse=False): Returns a new sorted list from the elements of an

iterable.

● range(stop), range(start, stop, step): Generates a sequence of numbers within a specified

range.

**2. User-Defined Functions:**

User-defined functions in Python allow you to create your own reusable blocks of code. Here

are the key aspects of defining user-defined functions in Python without specific code

examples:

Syntax: def *function\_name*(*parameters*):

*Function\_body*

● Function Name: Descriptive and follows naming conventions.

● Parameters: Variables in parentheses, used within the function.

● Function Body: Contains code executed when the function is called.

● Return Statement: Return specifies the value to be returned.

● Calling a Function: Executed by using function\_name(arguments).

● Default Values: Parameters can have default values.

Program

**3. Anonymous Functions:**

Anonymous functions in Python are created using the lambda keyword. Here are the key

aspects of anonymous functions:

Lambda Function: Anonymous functions are defined using the lambda keyword, followed by

parameters and an expression.

Syntax: lambda *parameters: expression.*

● No Name:

● Lambda functions are anonymous because they don't have a name like regular functions

defined with def.

● Single Expression:

● Lambda functions are limited to a single expression, and the result of the expression is

implicitly returned.

● Used for Small Operations:

● Lambda functions are often used for small operations and as arguments to higher-order

functions like map(), filter(), and sorted().

● Conciseness:

● Lambda functions are concise and useful when a full function definition is unnecessary.

● No Statements:

● Lambda functions can only contain expressions, not statements.

● Immutable:

● Lambda functions create anonymous functions, and once defined, their behavior cannot be

changed.

**Program:**

# Built-In Functions

print("eval function ->", eval("3+63-9"))

print("format function ->", format(0.085, '%'))

print("max function ->", max([23, 4, 5, 85, 96, 101]))

print("list & range function ->", list(range(9)))

print("type function ->", type(eval))

# User defined functions

def calculate\_square(x):

return x\*\*2

def calculate\_area(length, width):

return length \* width

def factorial(n):

if n == 0 or n == 1:

return 1

else:

return n \* factorial(n-1)

def fibonacci(n):

sequence = [0, 1]

while len(sequence) < n:

sequence.append(sequence[-1] + sequence[-2])

return sequence

print(f"Square using user-defined function: {calculate\_square(5)}")

print(f"Area using user-defined function: {calculate\_area(4, 2)}")

print(f"Factorial using user-defined function: {factorial(10)}")

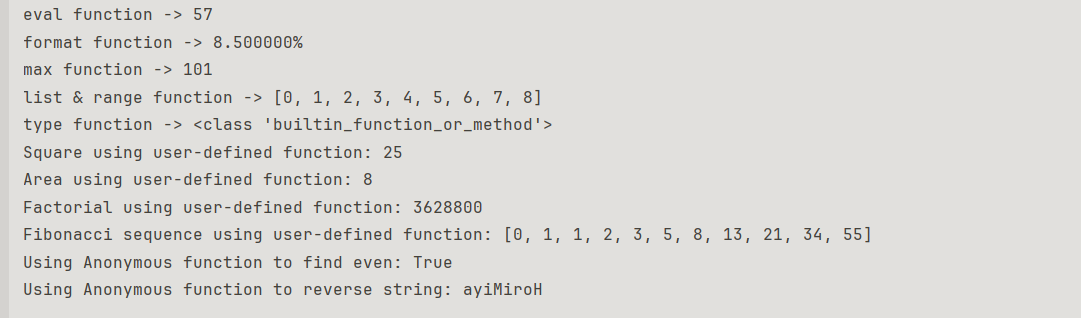
print(f"Fibonacci sequence using user-defined function: {fibonacci(11)}")

# Anonymous functions

print("Using Anonymous function to find even:", (lambda x: x % 2 == 0)(4))

print("Using Anonymous function to reverse string:", (lambda s: s[::-1])("HoriMiya"))

Output:



**Conclusion**: Different types of Functions are executed successfully in Python