



Submitted by	Fatima Shahzad BCS-22004
Submitted to	Sir Ghulam Mustafa
Course	Artificial Intelligence
Department	Information Technology
Discipline	Computer Science

LAB#2 OF ARTIFICIAL INTELLIGENCE

Iterative Structures in Python:

Python programming language provides following types of loops to handle looping requirements. Python provides three ways for executing the loops. While all the ways provide similar basic functionality, they differ in their syntax and condition checking time.

While Loop:

```
count = 0
while(count<3):
    count=count+1
    print("Python")
```

```
# Output:
# Python
# Python
# Python
```

Single statement while block:

```
count=0
while(count==0):
    print("Hello Geek")
#Output: infinite loop
```

for in Loop:

Example 1:

```
t=["Python","is", "interesting"]
for i in t:
    print(i)

# output:
# Python
# is
# interesting
```

Example 2:

```
print("\nTuple Iteration")
t = ("geeks", "for", "geeks")
for i in t:
    print(i)
# Output:
# Tuple Iteration
# geeks
# for
# geeks
```

Example 3:

```
s = "Geeks"
for i in s:
    print(i)
# Output:
# G
# e
# e
# k
# s
```

Iterating by index of sequences:

```
list = ["Python", "is", "Interesting"]
for index in range(len(list)):
    print(list[index])
#Output:
# Python
# is
# Interesting
```

Loop Control Statements:

Continue Statement:

```
for letter in 'programming':
    if letter == 'a' or letter == 'o':
        continue # Skip 'a' and 'o'
    print('Current Letter:', letter)

# Output:
# Current Letter: p
# Current Letter: r
# Current Letter: g
# Current Letter: r
# Current Letter: m
# Current Letter: m
# Current Letter: i
# Current Letter: n
# Current Letter: g
```

Break Statement:

```
for letter in 'examination':
    if letter == 'x' or letter == 'z':
        break # Loop will stop at 'x' or 'z'
    print('Current Letter:', letter)

# Output:
# Current Letter: e
```

Creating and Calling a Function:

```
def my_function():  
    print("Hello from a function")  
  
my_function()  
# Output:  
# Hello from a function
```

Passing a Parameters in Functions:

```
def my_function(fname):  
    print("Hello, " + name + "! Welcome.")  
  
my_function("Saba")  
my_function("Sana")  
my_function("Sara")  
  
# Output:  
# Hello, Saba! Welcome.  
# Hello, Sana! Welcome.  
# Hello, Sara! Welcome.
```

Default Parameter Value:

The following example shows how to use a default parameter value. If we call the function without parameter, it uses the default value:

```
def my_function(country="Norway"):  
    print("I am from " + country)  
  
my_function("Sweden")  
my_function("India")  
my_function()  
my_function("Brazil")
```

```
# output:  
# I am from Sweden  
# I am from India  
# I am from Norway  
# I am from Brazil
```

Passing a List as a Parameter:

Example:

```
def show_names(names):  
    for name in names:  
        print(name)  
  
friends = ["Saba", "Sara", "Sana", "Maryam"]  
show_names(friends)  
  
# Output:  
# Saba  
# Sara  
# Sana  
# Maryam
```

Return Values :

```
def my_function(x):  
    return 5 * x  
  
print(my_function(3))  
print(my_function(5))  
print(my_function(9))  
# Output:  
  
# 15  
# 25  
# 45
```

Keyword Arguments:

```
def my_function(child3, child2, child1):
    print("The youngest child is " + child3)

my_function(child1="Saba", child2="Sana", child3=Sara)
#Output:
#The youngest child is Sara
```

Python Classes/Objects:

Python is an object oriented programming language. Almost everything in Python is an object, with its properties and methods. A Class is like an object constructor, or a "blueprint" for creating objects.

```
Classes
class Myclass:
    x=5
p1=Myclass()
print(p1.x)

# output:
# 5
```

The init () Function:

All classes have a function called init (), which is always executed when the class is being initiated. Use the init () function to assign values to object properties, or other operations that are necessary to do when the object is being created

Example:

```
class Car:
    def __init__(self, brand, model, year):
        self.brand = brand
        self.model = model
```

```
        self.year = year

car1 = Car("Toyota", "Corolla", 2022)

print(car1.brand)
print(car1.model)
print(car1.year)

# Output:
# Toyota
# Corolla
# 2022
```

Object Methods:

```
class Car:
    def __init__(self, brand, model):
        self.brand = brand
        self.model = model

    def car_info(self):
        print("This car is a " + self.brand + " " + self.model)

car1 = Car("Toyota", "Corolla")

car1.car_info()

# output:
# This car is a Toyota Corolla
```