Department of
Computer Science and Engineering

Title: Python Programming (Loop, Functions, Class and Object)

Data Mining LAB

CSE 424

Green University of Bangladesh

1 Objective(s)

- To gather knowledge of Python program.

2 Python Programming

Python is an interpreted high-level general-purpose programming language. Python's design philosophy prioritizes code readability, as evidenced by its extensive use of indentation. Its language elements and object-oriented approach are aimed at assisting programmers in writing clear, logical code for both small and large-scale projects.

3 Loop

Most languages have the concept of loops: If we want to repeat a task twenty times, we don't want to have to type in the code twenty times, with maybe a slight change each time. As a result, we have for and while loops in python.

3.1 For Loop

Iterating over a sequence is done with a **for** loop (that is either a list, a tuple, a dictionary, a set, or a string).

3.1.1 Implementation in Python

```
# range(5) is not the values of 0 to 5, but the values 0 to 4.
for i in range(5):
  print(i)

-----------------------

# range(2, 5), which means values from 2 to 5 (but not including 5)
for i in range(2, 5):
  print(i)

-----------------------

# Increment the sequence with 3 (default is 1):
for i in range(2, 20, 3):
  print(i)

-----------------------

# Here, else keyword indicates a block of code to be executed when the loop is
    finished:
for i in range(5):
  print(i)
else:
  print("Finally finished!")

-----------------------

fruits = ["orange", "apple", "cherry"]
for i in fruits:
  print(i)

-----------------------

# Loop through the letters in the word "orange":
```

```
for i in "orange":
  print(i)

-----------------------

# Loop through the letters in the word "orange":

fruits = ["orange", "apple", "cherry"]
for i in fruits:
  print(i)
    if i == "apple":
        break
```

3.1.2 Lab Task (Please implement yourself and show the output to the instructor)

- Write a python program to find the sum of odd and even numbers from a set of numbers.
- Write a python program to find the smallest number from a set of numbers.
- Write a python program to find the sum of all numbers between 50 and 100, which are divisible by 3 and not divisible by 5.
- Write a python program to find the second highest number from a set of numbers.
- Write a python program to find the factorial of a number using for loop.
- Write a python program to generate Fibonacci series.

3.1.3 While Loop

3.1.4 Implementation in Python

```
i = 1
while i < 5:
  print(i)
  i += 1

-----------------------

i = 1
while i < 5:
  print(i)
  if i == 2:
    break
  i += 1

-----------------------

i = 0
while i < 5:
  i += 1
  if i == 2:
    continue
  print(i)

-----------------------

i = 1
while i < 5:
  print(i)
```

```
  i += 1
else:
  print("i is no longer less than 5")
```

3.1.5 Lab Task (Please implement yourself and show the output to the instructor)

- Write a python program to find the sum of odd and even numbers from a set of numbers.
- Write a python program to find the smallest number from a set of numbers.
- Write a python program to find the sum of all numbers between 50 and 100, which are divisible by 3 and not divisible by 5.
- Write a python program to find the second highest number from a set of numbers.
- Write a python program to find the factorial of a number using for loop.
- Write a python program to generate Fibonacci series.

4 Function

Functions enable you to break down the overall functionality of a script into smaller, logical subsections, which can then be called upon to perform their individual tasks when needed. Using functions to perform repetitive tasks is an excellent way to create code reuse. This is an important part of modern object-oriented programming principles.
In Python a function is defined using the **def** keyword:

4.1 Implementation in Python

```
def my_function():
  print("Hello World")

my_function()  # Calling my_function()
-----------------------

# Passing Argument
def my_function(name):
  print("name " + name)

my_function("Mike")
my_function("Monica")
my_function("John")

-----------------------

# Passing multiple Arguments
def my_function(fname, lname):
  print(fname + " " + lname)

my_function("Emil", "Refsnes")

-----------------------

# Arbitrary Arguments, *args
def my_function(*names):
  print("Name= " + names[2])

my_function("Mike", "Monica", "John")

```

```
-----------------------

# Send arguments with the key = value syntax.
def my_function(name3, name2, name1):
  print("Name= " + names3)

my_function(name1 = "Mike", name2 = "Monica", name3 = "John")

-----------------------

def my_function(**name):
  print("His last name is " + name["lname"])

my_function(fname = "Mike", lname = "Monica")

-----------------------

# Passing a List as an Argument
def my_function(food):
  for i in food:
    print(i)

fruits = ["orange", "apple", "cherry"]

my_function(fruits)

-----------------------

# Return Values
def my_function(a):
  return 5 * a

print(my_function(3))
print(my_function(5))
print(my_function(9))

-----------------------

# Recursion Example
def my_recursion(k):
  if(k > 0):
    result = k + my_recursion(k - 1)
    print(result)
  else:
    result = 0
  return result

print("\n\nRecursion Example Results")
my_recursion(6)
```

4.2 Lab Task (Please implement yourself and show the output to the instructor)

- Write a python program to find the largest number between two numbers using function
- Write a python program to find the sum of the numbers passed as parameters.

5 Classes and Objects

Python is a programming language that focuses on object-oriented programming. In Python, almost everything is an object, complete with properties and functions. A class functions similarly to an object constructor or a "blueprint" for constructing objects.

5.1 Create Class and Object

5.1.1 Implementation in Python

```
1 class MyClass:
2   a = 5
3
4 p1 = MyClass()
5 print(p1.a)
```

5.2 The _init_() Function

Every class has a method called _init_() that is invoked every time the class is started. To assign values to object properties or do other activities while the object is being constructed, use the _init_() function:

5.2.1 Implementation in Python

```
1 class Employee:
2   def __init__(self, name, age):
3     self.name = name
4     self.age = age
5
6 e1 = Employee("John", 26)
7
8 print(e1.name)
9 print(e1.age)
```

5.2.2 Input/Output

Output of the programs is given below.

```
John
26
```

5.3 Object Methods

Methods can also be found in objects. Objects have methods which are functions that belong to the object. To assign values to object properties or do other activities while the object is being constructed, use the _init_() function.

*The **self** parameter is a reference to the current instance of the class, and it is used to access class-specific variables. It doesn't have to be called self all the time; you can call it whatever you want, but it must be the first parameter of any class function*

5.3.1 Implementation in Python

```
1 class Employee:
2   def __init__(self, name, age):
3     self.name = name
4     self.age = age
```

```
5
6     def myfunc(self):
7       print("Hello my name is " + self.name)
8
9
10  e1 = Employee("John", 26)
11  e1.myfunc()
12
13  ------------------------------------------------
14
15  class Employee:
16    def __init__(myObject, name, age):
17      myObject.name = name
18      myObject.age = age
19
20    def myfunc(xy):
21      print("Hello my name is " + xy.name)
22
23
24  e1 = Employee("John", 26)
25  e1.myfunc()
```

5.3.2 Input/Output

Output of the programs is given below.

```
Hello my name is John
Hello my name is John
```

5.4 Modify and Delete Object Properties

5.4.1 Implementation in Python

```
1
2   # Modify Object Properties
3   class Employee:
4     def __init__(self, name, age):
5       self.name = name
6       self.age = age
7
8     def myfunc(self):
9       print("Hello my name is " + self.name)
10
11
12  e1 = Employee("John", 26)
13  e1.age = 30
14  print(e1.age)
15
16  ------------------------------------------------
17  # Delete Object Properties
18  class Employee:
19    def __init__(myObject, name, age):
20      myObject.name = name
21      myObject.age = age
22
23    def myfunc(xy):
24      print("Hello my name is " + xy.name)
25
```

```
26
27 e1 = Employee("John", 26)
28 del e1.age
29 print(e1.age)
```

5.4.2 Input/Output

Output of the programs is given below.

```
30
AttributeError: 'Employee' object has no attribute 'age'
```

5.4.3 Lab Task (Please implement yourself and show the output to the instructor)

- Define a Python function student(). Using function attributes display the names of all arguments.
- Write a Python function studentId () which will print the id of a student (studentId). If the user passes an argument studentId or studentId the function will print the student name and class.

6 Discussion Conclusion

Based on the focused objective(s) to understand about the python program, the additional lab exercise made me more confident towards the fulfilment of the objectives(s).

7 Lab Exercise (Submit as a report)

- Write a Python program to crate two empty classes, Student and Marks. Now create some instances and check whether they are instances of the said classes or not. Also, check whether the said classes are subclasses of the built-in object class or not.
- Write a Python class named Student with two attributes studentName, marks. Modify the attribute values of the said class and print the original and modified values of the said attributes.
- Write a Python class named Student with two attributes studentId, studentName. Add a new attribute studentClass and display the entire attribute and their values of the said class. Now remove the studentName attribute and display the entire attribute with values.
- Write a Python class named Student with two attributes studentId, studentName. Add a new attribute studentClass. Create a function to display the entire attribute and their values in Student class.

8 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.