BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

OBJECT ORIENTED DESIGN & PROGRAMMING (INSY 404) LECTURE SLIDES - 2

By

DR. EZE, M.O.

Department of Computer Science, Babcock University

Ogun State, Nigeria

BABCOCK UNIVERSITY

KNOWLEDGE
TRUTH
SERVICE

LECTURE MODULE 2

S/N	CONTENT
1.	CLASS CREATION
2.	OBJECT CREATION
3.	INSTANCIATION & METHODS
4.	SELF ARGUMENT
5.	TYPES OF CONSTRUCTORS
6.	__INIT__ METHOD
7.	QUESTION

CLASS CREATION

In order to proceed with OOP in Python, one must first understand how to create a class. In its simplest form, a Python Class can be created using the keyword **class** followed by the **name of class**, followed by **a colon**, then a **suite of statements.**

SYNTAX

```
class ClassName:
        #Statement Suite
```

BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

CLASS CREATION

IMPORTANT NOTE:

1. The keyword "**class**" should be in **lower case**, while the **class name** should start with a **capital letter**.
2. The statement suite (body of the class) consists of a number of statement types, ranging from **fields (properties)**, **constructors**, **functions**, **pass statements**, among others, as will be fully explained.
3. The body of the class starts on a new line, indented one tab from the left.
4. After creating a class, it has to be instantiated into objects.

BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

ILLUSTRATIONS

ILLUSTRATION 1:

```
class Employee:
    id = 10110
    name = “Sogun Dakwambo"
```

This illustration creates a Python class called **Employee**, which has two fields (id and name) for employee **id** and employee **name** respectively.

ILLUSTRATIONS

ILLUSTRATION 2:

```
class Dog:
    pass
```

This illustration creates a Python class called Dog. In this case, the statement suite is simply a **pass statement**. This is used in class definition to indicate that system should do nothing further.

BABCOCK UNIVERSITY

ILLUSTRATIONS

ILLUSTRATION 3:

```
# Creates class Car
class Car:

    # create class attributes
    name = "c200"
    make = "mercedez"
    model = 2010

    # create class methods
    def start(self):
        print ("Engine started")

    def stop(self):
        print ("Engine switched off")
```

In this illustration, we create a class named Car with three attributes: name, make, and model. The car class also contains two methods: start() and stop().

BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

OBJECT CREATION

Before a programmer can use a class, it has to be instantiated. It is through this process that an object gets created from a class. The syntax to create the instance of the class is as follows:

SYNTAX

```
ObjectName = ClassName (arguments)
```

OBJECT CREATION

IMPORTANT NOTE:

1. In the syntax statement for creating an object, there may or may not be arguments. In the absence of arguments, the bracket will be left empty. Example x = Marathon ( ).

ILLUSTRATIONS

ILLUSTRATION 4:

```
class Employee:
    id = 10
    name = "John"

emp = Employee()
emp.name="Samuel"
```

This illustration creates a Python object called emp from the class **Employee**. Note that emp object inherited the fields (attributes) of the Employee class. Thus, we have an assignment statement using emp.name.

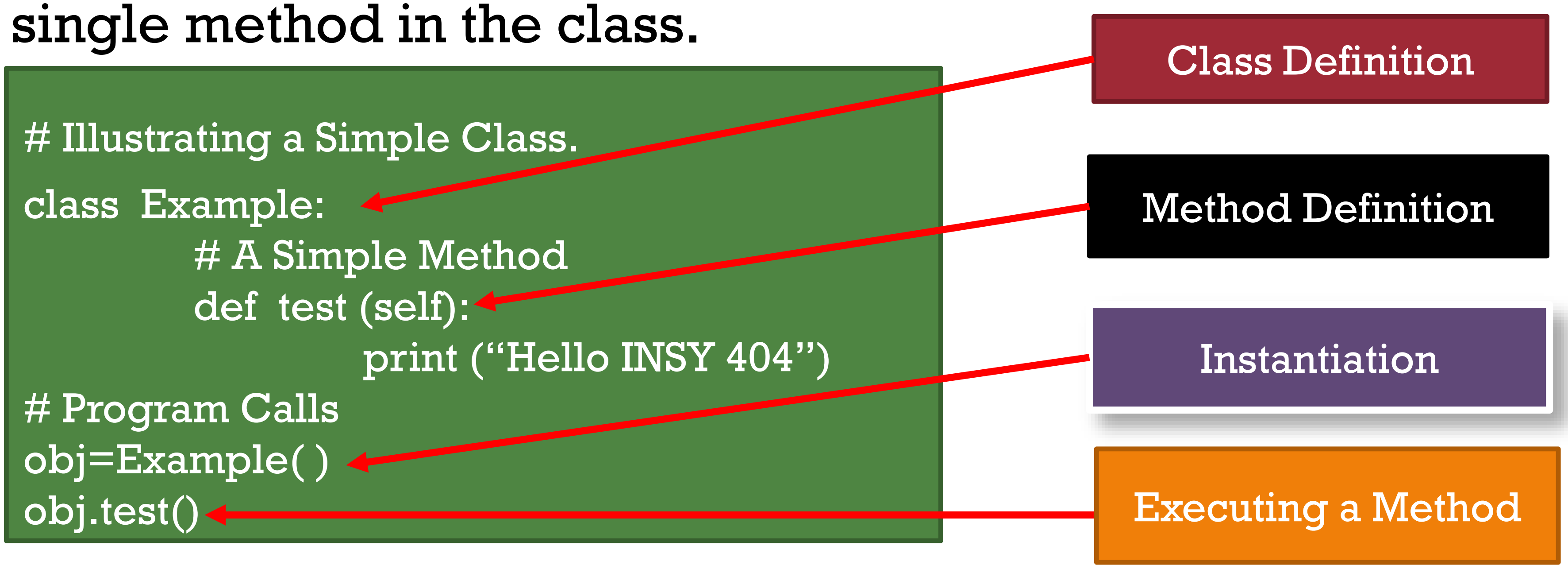
```
# A sample method
def fun(self):
    print("Hello")

# Driver code
obj = Test()
obj.fun()
```

INSTANTIATION & METHOD

ILLUSTRATION 5:

The following code shows a given class with a single method. for a given class. Here we created a new class called Example. This is followed by an indented block of statements which form the body of the class. In this case, we have defined a single method in the class.



```
# Illustrating a Simple Class.
class  Example:
        # A Simple Method
        def  test (self):
                print (“Hello INSY 404”)
# Program Calls
obj=Example( )
obj.test()
```

BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

THE SELF ARGUMENT

As shown in the last illustration:

1. Class methods must have an extra first parameter (known as **self**) in the method definition.
2. In real life, the number of parameters in a function definition must correspond with number of arguments in function calls. But this is defied in Python.

REAL LIFE FUNCTIONS:	**PYTHON METHODS**
Function Definition: MathF(Para1, Para2,…ParaN)	**Method Definition:** PythM(Self, Para2,…ParaN)
Function Call: X=MathF(Arg1, Arg2,…ArgN)	**Method Call:** X=PythM (Arg2,…ArgN)

3. The first parameter is not given value (arguments) during method calls. It is usually provided internally by Python.

THE SELF ARGUMENT

NOTE:

1. Even if a method takes no argument during its call, we still have to insert one parameter - the **self** – as shown in the last illustration **test( ).**
2. This is similar to **this** pointer in **C++** and **this** reference in **Java.**
3. When you call a method of the object as myobj.method (arg1, arg2), this is automatically converted by Python into MyClass.method(myobj, arg1, arg2).
4. This is the **self** special parameter in Python.

BABCOCK UNIVERSITY
KNOWLEDGE TRUTH SERVICE

THE CONSTRUCTOR

A constructor is defined as a special method used for initializing the instance variables during object creation. A Python Constructors is usually implemented using __init__ (self). There are three major types constructors (though some text books usually recognize only two – numbers 2 and3 below), all to be explained in details at a later section:

1. Default Constructor
2. Non-Parameterized Constructor
3. Parameterized Constructor

BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

THE INIT METHOD

In Python, Constructors are implemented using __init__ method. It is run as soon as an object of a class is instantiated.

```
# A Sample class with init method (demo_init.py)
class Person:

    # init method or constructor
    def __init__(self, name):
        self.name = name

    # Sample Method
    def greet_you(self):
        print(‘Hello, my name is’, self.name)

p = Person(‘Salako’)
p.greet_you ()
```

BABCOCK UNIVERSITY

KNOWLEDGE
TRUTH
SERVICE

CLASS ASSIGNMENT

1. ***A Book has four chapters, with number of pages as 20, 10, 17 and 21 respectively. Create a Python class known as Book, with four fields as number of pages in each of the chapters.***