BABCOCK UNIVERSITY

KNOWLEDGE
TRUTH
SERVICE

OBJECT ORIENTED DESIGN & PROGRAMMING (INSY 404) LECTURE SLIDES - 1

By

DR. EZE, M.O.
Department of Computer Science, Babcock University
Ogun State, Nigeria

BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

LECTURE MODULE 1

IN THE BEGINNING

Though OOP is considered to be a modern programming paradigm, the root goes back to **1960s.** The first programming language to use objects was **Simula 67**, which was introduced in the year **1967.** A major breakthrough for OOP came with the programming language **Smalltalk in the 1970s.**

BABCOCK UNIVERSITY

KNOWLEDGE
TRUTH
SERVICE

PROGRAMMING PARADIGMS

Different programming languages have their individual areas of strength and focus. Similarly, their usage may vary drastically, depending on the type of problems being solved. Eg. It is better to cut a tree with a sharp axe than a sharp razor. But to trim a piece of paper, one should use a sharp razor rather than an axe! A study of "***Survey in Programming Languages***" helps students to understand the strengths and weaknesses of different languages, and the choice of which paradigm fits which problems. Apart from OOP, other programming paradigms exist, as shown in the next slide.

PROGRAMMING PARADIGMS

1. Imperative Programming (Sequence of Operations)
2. Object Oriented Programming
3. Concurrent Programming (Parallel/Real Time Systems)
4. Functional Programming (Mainly uses Functions)
5. Logic Programming (Used in Expert Systems Development)
6. Event-Driven Programmimg

WHAT IS OOP?

One of the popular approaches to solving a programming problem is by creating objects. This is known as Object-Oriented Programming (OOP).

An object is a thing or idea that you want to model in your program. An object can be anything, tangible or intangible.

Examples: Employee, Bank account, Car, Air, etc.

BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

PROGRAMMING TOOLS

In this OOP course, we shall be using Python as a Programming Tool. Python is a **multi-paradigm programming language**. This means, that it supports different programming approaches as listed in the earlier slide. The concept of OOP in Python focuses on creating reusable code. This concept is also known as DRY (**Don't Repeat Yourself**).

BABCOCK UNIVERSITY

KNOWLEDGE
TRUTH
SERVICE

OOP KEY CONCEPTS

OOP CONCEPTS

1. CLASS
2. OBJECT
3. POLYMORPHISM
4. INHERITANCE
5. ENCAPSULATION
6. ABSTRACTION

DEFINITIONS

CLASS: A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type.

BABCOCK UNIVERSITY

KNOWLEDGE
TRUTH
SERVICE

DEFINITIONS

OBJECT: This is a basic unit of Object Oriented Programming and represents the real life entities. A typical OOP program creates many objects, which interact by invoking methods. An object consists of:

- **Attributes** (The properties of an object)
- **Behaviours** (The responses/actions of an object especially in relation with other objects)

BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

DEFINITIONS

INHERITANCE: Inheritance is an important aspect of the OOP. Inheritance provides code reusability, because it enables the programmer to use an existing class to create a new class instead of creating it from scratch.

Through Inheritance, a **Child class** conveniently acquires the properties and can access all the **data members** and **functions** defined in the **Parent class.** Moreover, a **Child class** can also provide its specific implementation to the functions of the **Parent class**.

BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

DEFINITIONS

POLYMORPHISM: This refers to the ability of a system to take various forms. Polymorphism in Python allows a programmer to define methods in the child class with the same name as defined in their parent class. A child class inherits all the methods from the parent class. However, there are cases where the method inherited from the parent class doesn't quite fit into the child class. In such cases, a programmer will have to re-implement method in the child class. This process is known as **Method Overriding**.

BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

DEFINITIONS

ENCAPSULATION: Encapsulation is the wrapping up of data under a single unit. This mechanism binds together code and the data it manipulates. Another way to think about encapsulation is, that it is "***a protective shield that prevents the data from being accessed by the code outside this shield***". Technically in encapsulation, the variables or data of a class is hidden from any other class and can be accessed only through any member function of own class in which they are declared. Through encapsulation, the data in a class is hidden from other classes, so it is also known as **data-hiding.**

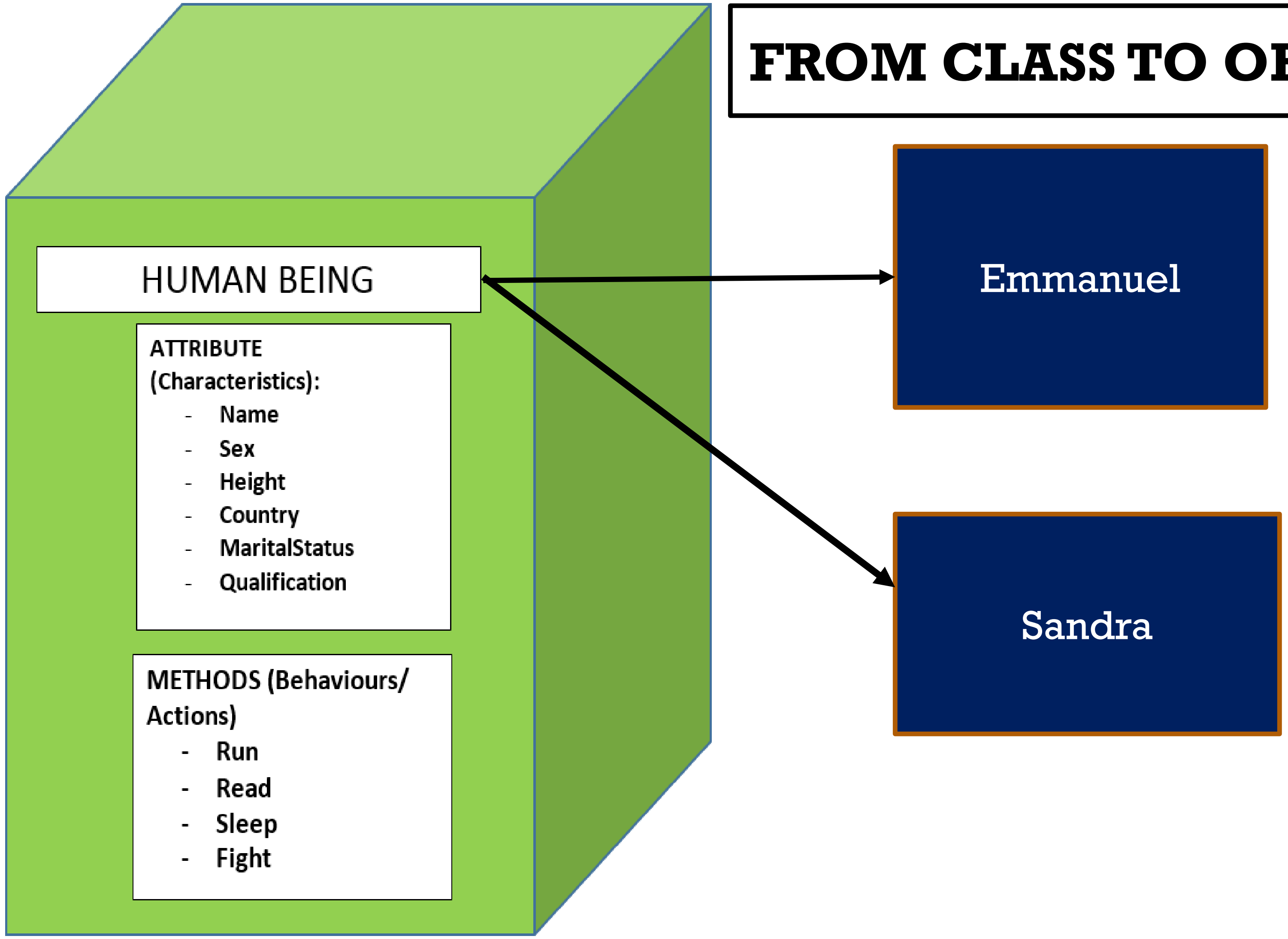
BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

DEFINITIONS

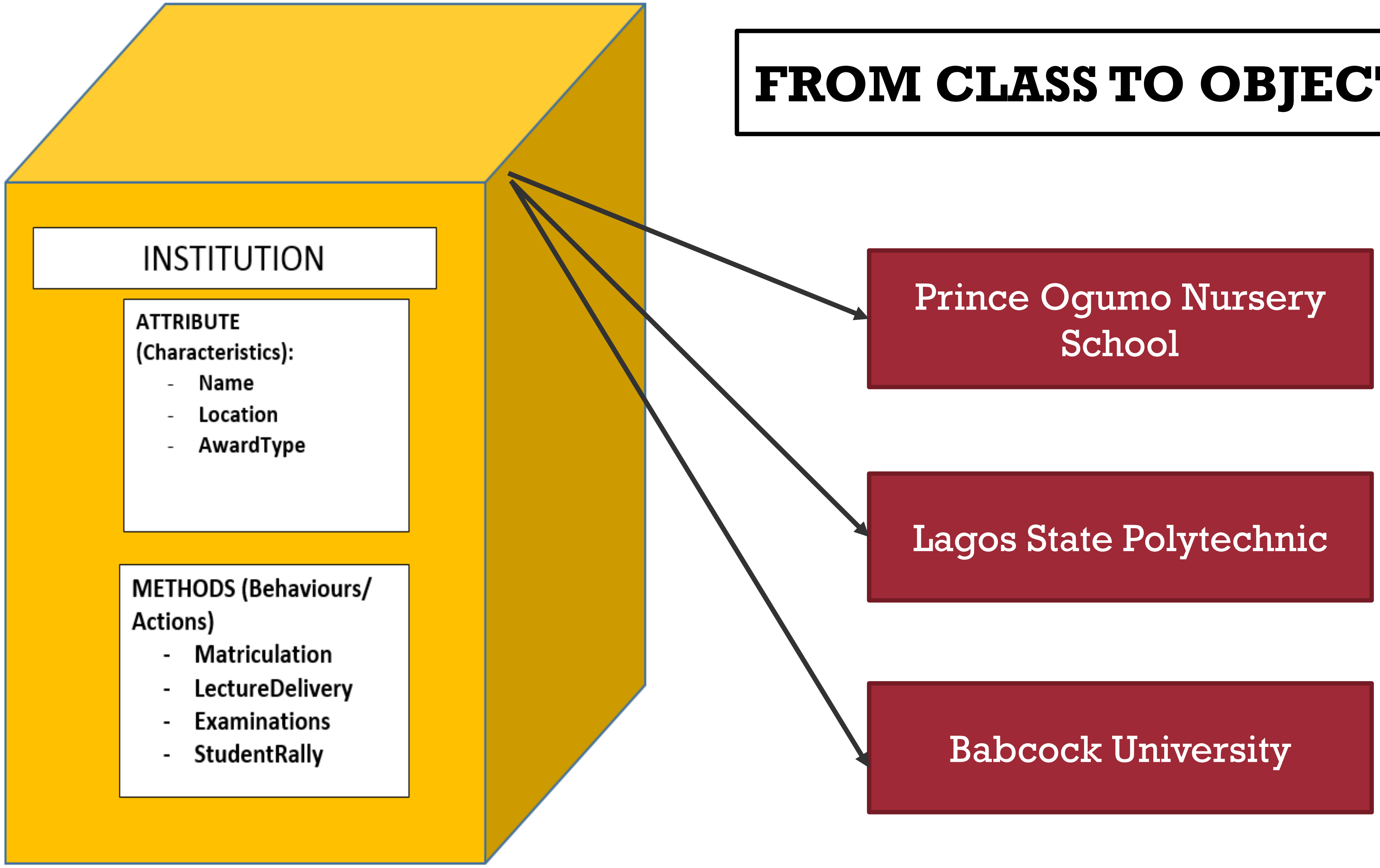
DATA ABSTRATION: Data Abstraction is the property by virtue of which only the essential details are displayed to the user. The trivial or the non-essentials units are not displayed to the user. Data Abstraction may also be defined as "***the process of identifying only the required characteristics of an object, while ignoring the irrelevant details***". Example: Consider a non-computer literate old man carrying a laptop. He is aware that by pressing the ON button, the laptop boots. Thus, even though he does not know (and may never care to know) how the CPU and other internal components actually work, he can conveniently power on the machine.

DIAGRAMATIC ILLUSTRATIONS



DIAGRAMATIC ILLUSTRATIONS

FROM CLASS TO OBJECTS



BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

COMPARATIVE ANALYSIS

ADVANTAGES OF OOP:

1. Object-oriented programming encourages reusability. Programs written in the form of objects and classes, can easily be reused in other projects.
2. The source code resulting from object-oriented programming is highly maintainable due to the modular approach applied.
3. Program correctness could be enhanced in OOP since every class has a specific task. Thus, an error in one part of the code, could be rectified locally without having to affect other parts of the code.

BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

COMPARATIVE ANALYSIS

ADVANTAGES OF OOP:

4. Data encapsulation adds an extra layer of security to the program developed using the object-oriented approach.

DISADVANTAGES OF OOP:

1. In OOP, the program designer needs to create the requisite objects. Creating objects require detailed domain knowledge of the software being developed. This initial task could be hard for newbies.

BABCOCK UNIVERSITY

KNOWLEDGE TRUTH SERVICE

COMPARATIVE ANALYSIS

DISADVANTAGES OF OOP:

2. The size and complexity of the program grows exponentially as the programmer adds more and more classes to the code. Thus, the task of debugging could become more difficult.

CLASS ASSIGNMENT

1. ***Explain the difference between OOP in Python and in Java.***