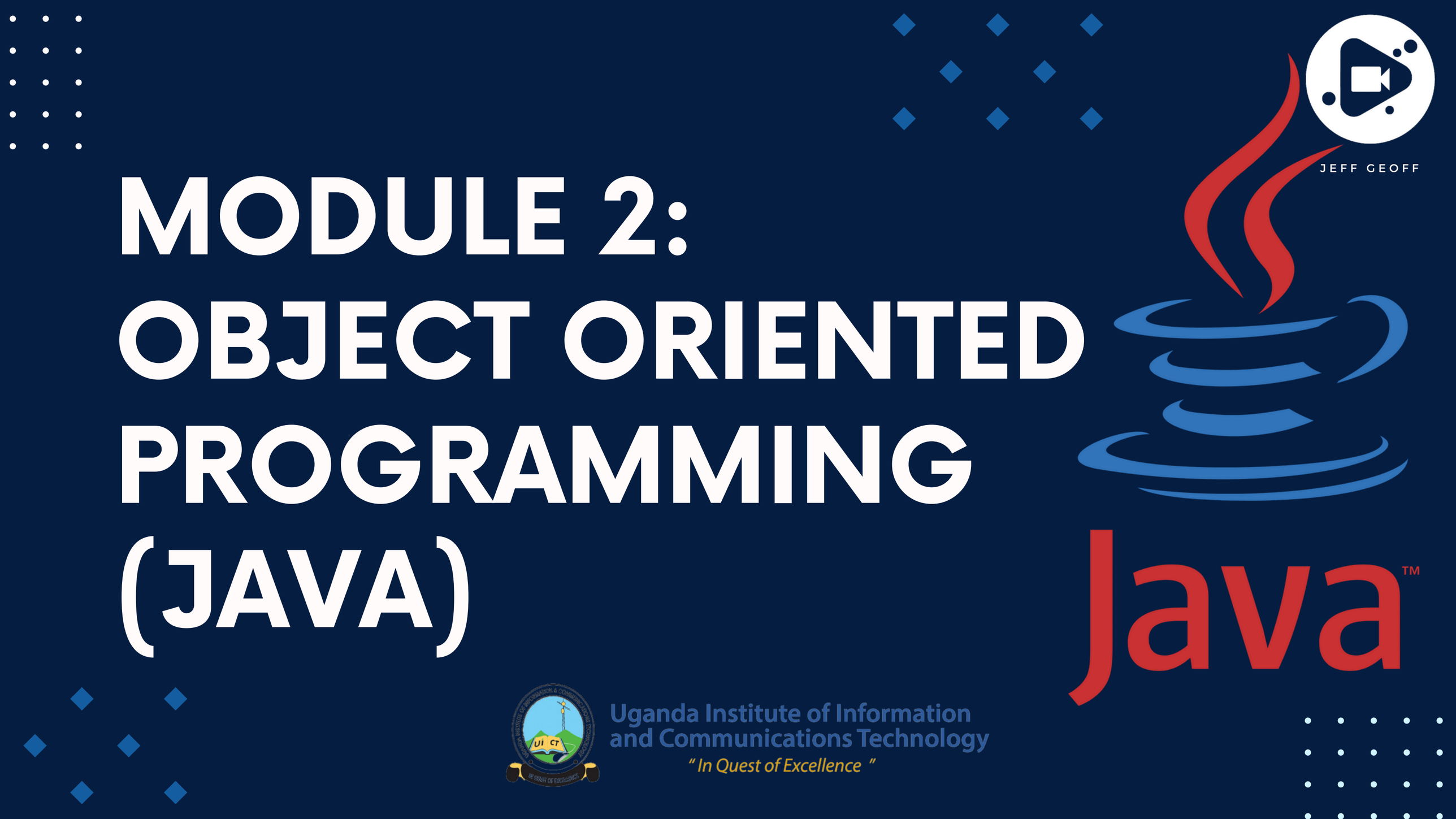
JEFF GEOFF
MODULE 2:
OBJECT ORIENTED
PROGRAMMING
(JAVA)
Java™
Uganda Institute of Information
and Communications Technology
" In Quest of Excellence "

MODULE 2: OBJECT-ORIENTED PROGRAMMING

- Classes and objects
- Inheritance
- Polymorphism
- Encapsulation
- Abstraction

Uganda Institute of Information and Communications Technology

"In Quest of Excellence"

Connect with me: **@Jeff_Geoff_Masterclass**

Module 2: Object-Oriented Programming

Object-oriented programming (OOP) is a programming paradigm that uses objects and their interactions to design applications and computer programs. OOP is one of the most popular programming paradigms today, and it is used to develop a wide variety of applications, including web applications, mobile apps, desktop applications, and enterprise software.

OOP IS BASED ON THE FOLLOWING FOUR PRINCIPLES:

- Abstraction: Abstraction is the process of hiding unnecessary details and exposing only the essential information about an object. This makes it easier to understand and use objects.

- Encapsulation: Encapsulation is the process of bundling together an object's data and behavior into a single unit. This makes objects self-contained and easier to manage.

The link will take you to the course’s YouTube channel

OOP IS BASED ON THE FOLLOWING FOUR PRINCIPLES:

- Inheritance: Inheritance is the process of creating a new object from an existing object. This allows you to reuse code and avoid writing the same code over and over again.

- Polymorphism: Polymorphism is the ability of an object to take on different forms. This makes objects more flexible and easier to use in different situations.

The link will take you to the course’s YouTube channel

OOP provides a number of benefits, including:

- **Modularity: OOP makes it easy to break down complex problems into smaller, more manageable pieces. This makes code easier to write, maintain, and debug.**
- **Reusability: OOP allows you to reuse code by creating new objects from existing objects. This saves time and effort, and it also helps to improve the quality of your code.**
- **Maintainability: OOP makes code easier to maintain by encapsulating objects and hiding unnecessary details. This makes it easier to understand and modify code without affecting other parts of the program.**

Classes and objects

In Java, classes and objects are fundamental concepts used for creating reusable code and modeling real-world entities.

Example 1

Suppose we want to model a simple Person entity with a name and age

Live Demo with VS Code

Example 2

Suppose we create a BankAccount class to represent a basic bank account and demonstrate creating multiple objects from the class:

Live Demo with VS Code

Inheritance

Inheritance is a fundamental concept in object-oriented programming (OOP) that allows you to create new classes (derived or child classes) based on existing classes (base or parent classes).

Inheritance ...

In Java, you can achieve inheritance using the extends keyword.

Example 1: Inheriting from a Parent Class

Suppose we have a parent class Vehicle and want to create child classes Car and Bicycle that inherit properties and methods from the parent class.

Live Demo with VS Code

Example 2: Multilevel Inheritance

In multilevel inheritance, a child class becomes the parent class for another child class

Polymorphism

Polymorphism is a fundamental concept in object-oriented programming (OOP) that allows objects of different classes to be treated as objects of a common superclass

In Java, polymorphism is primarily achieved through method overriding and interfaces

Example 1

In this example, we'll demonstrate polymorphism using method overriding. We have a superclass Shape with a method calculateArea(), and two subclasses Rectangle and Circle that override this method to calculate the area of their respective shapes.

Live Demo with VS Code

Encapsulation

It refers to the concept of bundling data (attributes) and methods (functions) that operate on that data into a single unit, known as a class.

Encapsulation restricts direct access to some of an object's components, providing control over the object's state and behavior.

Example 2

In this example, we'll create a Person class with private attributes (name, age, and email) and public getter and setter methods to access and modify these attributes in a controlled manner:

Live Demo with VS Code

Abstraction

Abstraction is one of the four fundamental principles of object-oriented programming (OOP) that focuses on simplifying complex systems by modeling classes based on their essential characteristics while hiding unnecessary details.

Abstraction ...

Abstraction allows you to define a blueprint (class) that represents the common attributes and behaviors of objects, without specifying every detail

Example 3

In this example, we'll create an abstract class Shape that defines the common attributes and a method area() for calculating the area of various shapes. We'll also create concrete subclasses, Circle and Rectangle, that implement the area() method according to their specific shapes:

Live Demo with VS Code

Thank's For Listening

Connect with us.

+250 722 359 866

jeff.geoff.phd@gmail.com

OBJECT ORIENTED PROGRAMMING (JAVA)

Java™