# 20CYS312 - Principles of Programming Languages
# Exploring Programming Paradigms

**Assignment-01**

**Presented by Saranesh E S**
**CB.EN.U4CYS21068**
**TIFAC-CORE in Cyber Security**
**Amrita Vishwa Vidyapeetham, Coimbatore Campus**

Feb 2024


AMRITA
VISHWA VIDYAPEETHAM

# Outline



**Principles of Object-Oriented Programming:**

1. Inheritance: The capability of a class to derive properties and characteristics from another class is called Inheritance.
2. Encapsulation: Encapsulation is the concept of binding fields and methods together as a single unit. A class allows programmers to create objects with variables (data) and behaviors (methods or functions).
3. Abstraction: Abstraction is a methodology in which details of the programming codes are hidden away from the user, and only the essential things are displayed to the user.
4. Polymorphism: Polymorphism is the ability of any data to be processed in more than one form. It describes the concept that you can access objects of different types through the same interface.

**Applications:**

1. Hypertext and Hypermedia: Hypertext is similar to regular text as it can be stored, searched, and edited easily. Hypermedia on the other hand is a superset of hypertext.
2. Client-Server systems: OOPs principles are quite helpful. To construct Object-oriented server internet, or OCSI, applications, the IT infrastructure is created using Object-oriented client-server systems.
3. Object-oriented Database: To preserve the object's integrity and identity, databases maintain a close correlation between database objects and their counterparts in the physical world.
4. . Computer Aided Designs: OOP can be used in CAD software to develop classes that represent different design components including lines, curves, surfaces, and solids.

**Features of PHP:**

1. Simplicity: Easy to learn the language as the syntax is similar to that of 'C' or Pascal language. language is very logical and well organized generalpurpose programming language.
2. Objective oriented: PHP supports object-oriented programming features like data encapsulation, inheritance, abstraction, polymorphism, etc.
3. Open Source: All PHP frameworks are open sources, No payment is required for the users and its completely free.
4. Error reporting and exceptions: PHP has many pre-defined functions and reporting constants that generate errors at runtime. There are 16 levels of error in PHP5, representing the category and severity of an error in PHP.

**Features of PHP:**

1. Object oriented features: PHP supports object-oriented programming features, resulting in increased speed and introducing added features like data encapsulation and inheritance at many levels.
2. Real time access monitoring: PHP also provides a summary of user's recent logging accesses.
3. Flexibility: it can be well integrated with HTML, XML, Javascript and many more. PHP can run on multiple operating systems like Windows, Unix, Mac OS, Linux, etc. It is very comfortably integrated with various Databases.

# Imperative

**Characteristics of Imperative Programming:**

1. Procedure call: The effect of a procedure call is to apply a procedure abstraction to some arguments. The net effect of a procedure call is to update variables (local or global).
2. Sequential commands: Much of imperative languages are concerned with control flow, making sure that commands are executed in a specific order.
3. Collateral commands: A computation is deterministic if we can predict in advance exactly which sequence of steps will be followed. Otherwise the sequence is nondeterministic.
4. Conditional commands: A conditional command has a number of subcommands, from which exactly one is chosen to be executed.
5. Iterative commands: An iterative command, also known as loop, has a set of commands that is to be executed repeatedly and some kind o phrases that determines when the iteration will stop.

## Concepts of Imperative Programming:

1. State Modification:Imperative programming involves statements that changea program's state. Programs consist of commands for the computer to perform, much like imperative mood in natural languages expresses commands.
2. Procedural Programming: Procedural programming is a type of imperative programming. Programs are built from procedures (subroutines or functions).
3. Structured Programming: Heavy procedural programming, where state changes are localized to procedures, is a form of structured programming. Structured programming techniques improve maintainability and overall quality of imperative programs.
4. Object-Oriented Programming (OOP): Object-oriented programming extends structured programming concepts.

## Key COBOL features

1. Global business language: Business-Centric Design, Early Adoption in Business Applications, Stability and Reliability were some reasons why it was a global business language.
2. Seamless integration with modern systems: COBOL is a legacy language that supports and integrates easily with most traditional deployments, architectures, modern technologies, and complex applications.
3. Easy readability: COBOL came into existence with the motto of developing a language that communicates better with computers. The language does not use pointers, user-defined data types or functions, making it a simple language to understand
4. Portable language: COBOL programs run on different platforms. The platform-agnostic aspect also allows developers to build, test, and deploy COBOL programs across various supported platforms, thereby speeding up the development and application execution process.

## Application of Cobol

1. Financial Systems: widely used in the financial industry for applications such as banking systems, accounting software, and financial transaction processing.
2. Insurance Applications: used for policy management, claims processing, and other critical functions. COBOL's robustness and ability to handle large amounts of data make it suitable for insurance-related applications
3. Government Systems: used for applications like tax processing, social security systems, and public administration. The language's stability and longevity make it a reliable choice for maintaining essential government services.
4. Healthcare Systems: used for managing patient records, billing systems, and other administrative tasks. The language's suitability for processing and organizing large datasets is beneficial in healthcare applications.

# Comparison

**Comparison of the two paradigms and languages - Similarities:**

1. Programming Paradigm
2. Platform Independence
3. Business Applications
4. Data Storage and File Handling:
   - File Handling
   - Data Storage

**Comparison of the two paradigms and languages - Differences:**

1. Domain and Usage
2. Web vs. Mainframe
3. Syntax and Structure
4. Community and Ecosystem

# Conclusion

1. The choice between OOP and Imperative Programming depends on the specific needs of the application and the preferences of the development team. OOP brings modularity, code reuse, and abstraction, making it suitable for complex systems.
2. Imperative Programming, exemplified by COBOL, shines in data-centric, transactional applications where efficiency and reliability are paramount.
3. Understanding the strengths and weaknesses of each paradigm allows developers to make informed decisions based on the requirements of the project. Ultimately, both paradigms have their place in the diverse landscape of programming languages, each addressing unique challenges and catering to distinct application domains.

# Bibliography


- https://www.educative.io/blog/object-oriented-programming
- https://www.ibm.com/docs/en/watsonx-as-a-service?topic=language-object-oriented-programming
- https://www.javatpoint.com/inheritance
- https://www.scaler.com/topics/encapsulation-in-oops/
- https://www.analyticsinsight.net/top-10-applications-of-object-oriented-programming-using-python-language/
- https://www.php.net/manual/en/index.php
- https://pbphpsolutions.com/oops-concepts-in-php-with-realtime-examples.html
- https://www2.seas.gwu.edu/