OBJECT ORIENTED PROGRAMMING

Branch: S6 ECE

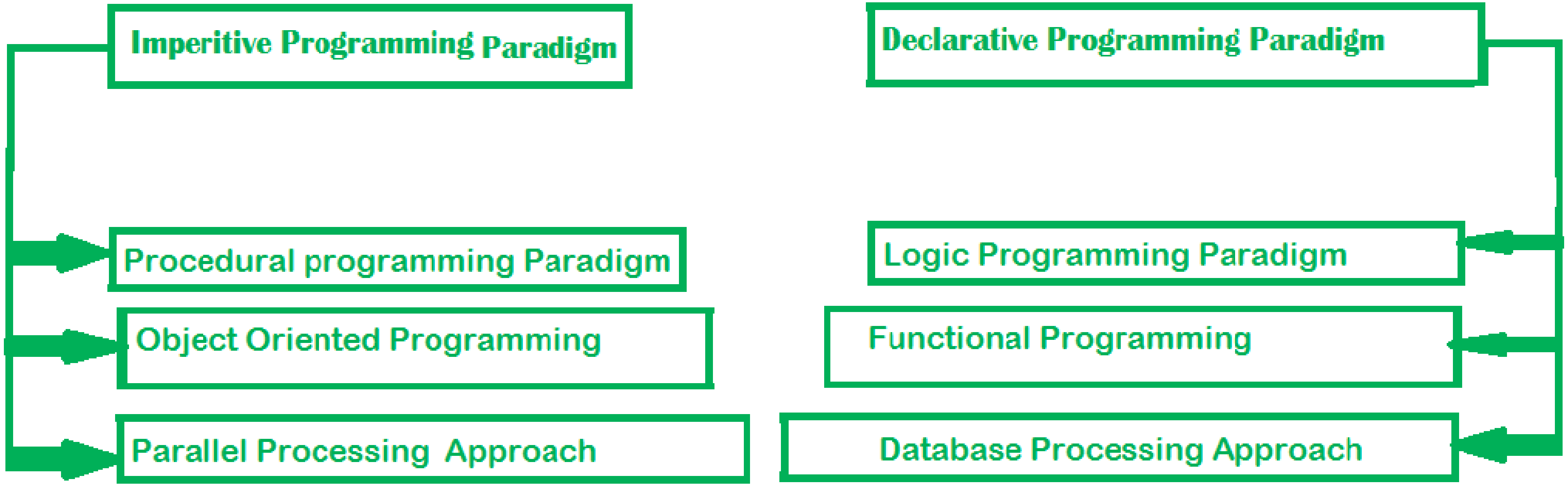
Faculty: SREEDIVYA I

PRINCIPLES OF OOP

PROGRAMMING PARADIGMS

- Paradigm-method to solve a problem
- Programming Paradigm- method to solve problem using some programming language

Programming Paradigms



Imperitive Programming Paradigm

Procedural programming Paradigm

Object Oriented Programming

Parallel Processing Approach

Declarative Programming Paradigm

Logic Programming Paradigm

Functional Programming

Database Processing Approach

A. Imperative Programming Paradigm

- Oldest programming paradigm
- Its features are closely related to machine language
- Based on Von Neumann architecture
- Assignment statements are used for changing program state
- Performs step by step task by changing state
- Three categories:

1. Procedural Programming
2. Object oriented Programming
3. Parallel Processing Approach

Advantages

- Very simple to implement
- It contains loops, variables etc.

Disadvantages

- Complex problem cannot be solved
- Less efficient and less productive
- Parallel programming is not possible

1.Procedural Programming Paradigm

- Emphasize on procedures i.e., how to do a particular task rather than what to do
- Ability to reuse the code
- Examples: C, Fortran, Pascal

2.Object Oriented Programming

- Consists of classes and objects
- More emphasis on data
- Able to solve all real world problems
- Examples: C++, Java, Python

Advantages

- Data security
- Inheritance
- Code reusability
- Flexible and abstraction is also present

3.Parallel Processing Approach

- Parallel processing is the processing of program instructions by dividing them among multiple processors.
- Objective - running a program in less time by dividing them
- Example - NESL

B. Declarative Programming Paradigm

- It expresses logic of computation without mentioning its control flow
- Considers programs as theories of some logic
- Emphasis is on what to do and just declares the result
- Three categories

1. Logic
2. Functional
3. Database Processing

1.Logic Programming

- abstract model of computation
- solve logical problems like puzzles, series
- Here we have a knowledge base which we know before and along with the question and knowledge base which is given to machine, it produces result.
- Emphasize is on knowledge base and the problem
- The execution of the program is very much like proof of mathematical statement
- Example: Prolog

2.Functional Programming

- The key principle is the execution of series of mathematical functions
- Functions hide the implementation details
- Function can be replaced with their values without changing the meaning of the program
- Example: Perl, JavaScript

3.Database Programming

- Based on data and its movement
- Program statements are defined by data
- Provides file creation, data entry, update, query and reporting functions
- Example: SQL

Difference between POP and OOP

Procedure Oriented Programming	Object Oriented Programming
Follows Top down approach	Follows Bottom up approach
Programs are divided into small parts called functions	Programs are divided into small parts called objects
Data is less secure	Data is highly secured
Deals with algorithm	Deals with data
Very less memory is required	Memory requirement is more than POP
No access specifier	Has Access specifier(public,private,protected)
Cannot perform overloading	Can perform overloading
Data hiding is not possible	Data hiding is possible
Examples: C, Fortran, Cobol	Examples: C++, Java, Python