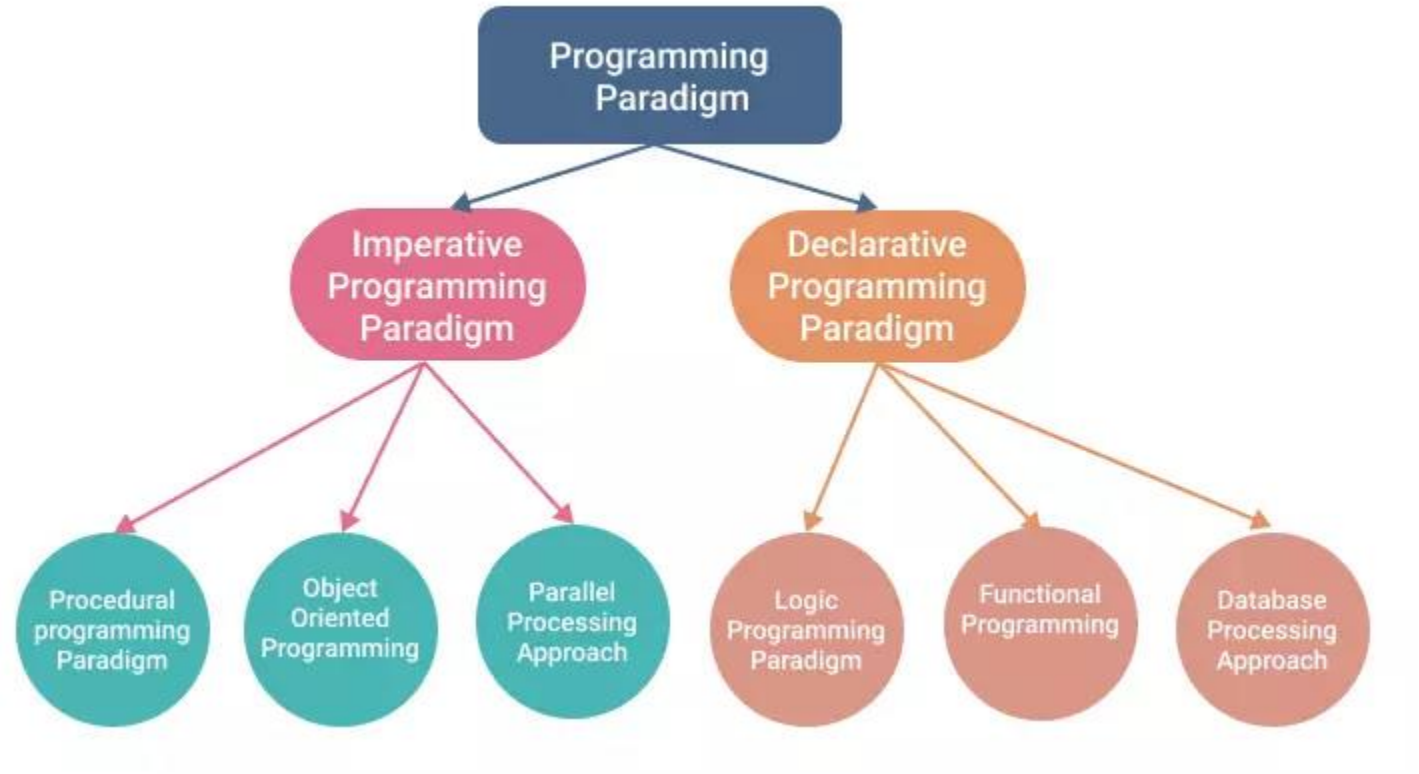
Programming Paradigms



Paradigm is a method to solve some problem or do some task. Programming paradigm is an approach to solve a problem using some programming language or also we can say it is a method to solve a problem using tools and techniques that are available to us following some approach.

1. Imperative programming paradigm:

It performs step by step task by changing state. The main focus is on how to achieve the goal. The paradigm consist of several statements and after execution of all the result is stored.

A- Procedural programming:

step-by-step approach, where programs are divided into procedures (also called functions or routines). It follows a top-down approach, focusing on a sequence of instructions that operate on data.

```cpp
int main() {
    // Array to store marks
    int marks[5] = { 12, 32, 45, 13, 19 };

    // Variable to store the sum of marks
    int sum = 0;

    // Variable to store the average
    float average = 0.0;

    // Calculate the sum of marks
    for (int i = 0; i < 5; i++) {
        sum = sum + marks[i];
    }

    // Calculate the average
    average = sum / 5.0;

    // Output the average
    std::cout << "Average of five numbers: " << average << std::endl;

    return 0;
}
```

B- Object oriented programming

It works based on the concept of objects, which encapsulate data (attributes) and behavior (methods). It follows principles such as encapsulation, inheritance, polymorphism, and abstraction. OOP promotes reusability and modularity.

```
class Car
{
    // Attributes (fields)
    public string Brand; public string Model; public int Year;
    // Constructor
    1 reference
    public Car(string brand, string model, int year)
    { Brand = brand; Model = model; Year = year;  }
   // Method to display car details
    1 reference
    public void DisplayInfo() {Console.WriteLine($"Car: {Brand} {Model}, Year: {Year}");}
}
0 references
class Program
    0 references
{   static void Main()
    {    // Creating an object of Car
        Car myCar = new Car("Toyota", "Corolla", 2022);
        myCar.DisplayInfo(); // Output: Car: Toyota Corolla, Year: 2022
    }
}
```

C- Parallel processing

A type of computation where multiple processes or threads execute simultaneously to improve performance and efficiency.

---

2. Declarative programming paradigm:

It's a style of building programs that expresses logic of computation without talking about its control flow, focuses on **what** the outcome should be rather than **how** to achieve it.

A- Logic programming:

It's based on formal logic, where a program consists of a set of facts and rules, and execution involves querying the system to infer conclusions using logical deduction.
commonly used in artificial intelligence, knowledge representation, and expert systems.
Such as: Prolog

B- Functional programming

It treats computation as the evaluation of mathematical functions, avoiding changing state and mutable data. It emphasizes **pure functions, immutability, and higher-order functions**.
commonly used in data processing and mathematical computations.
Such as Haskel

C- Database/Data driven programming

A programming approach where the flow and logic of the application are determined by data stored in a database rather than by hardcoded logic.
commonly used in web applications, business software, and enterprise systems where structured data management is essential.
Such as SQL