COMSATS UNIVERSITY, ISLAMABAD



**Programming Fundamentals**

--CSC103--

**Name**: Talha Rizwan

**Reg no**. SP20-BSE-093

**Section**: B

**Submitted to**: Dr. Behjat Zuhaira

*Date*: 9/26/2020

# Programming Paradigm:

A programming paradigm is a philosophy, style, or general approach to writing code. It does not refer to a specific language. It refers to the way how you program. In today’s world there are a lot of programming languages but all of them are following some strategies and rules. And those strategies and rules are known as Programming Paradigms.

# Classification of Programming Paradigms:

There are two major Paradigms. Declarative and Imperative. All the other types are derived from these two basic types. Following is the diagram of classification of Programing Paradigms.

Functional reactive programming (FRP)

Concurrent logic programming

Nonmonotonic dataflow programming

Multi-agent dataflow programming

Object-capability programming

Active object programming

Message-passing concurrent programming

Multi-agent programming

Event loop programming

Encapsulated functional programming

Software transactional memory (STM)

Shared-State concurrent programming

Concurrent object-oriented programming

Stateful functional programming

Sequential object-oriented programming

Lazy declarative concurrent programming

Lazy dataflow programing

Lazy Constraint programming

Concurrent constraint programming

Constraint Programming

Relational and logical programming

Monotonic data flow programing

Lazy functional programming

Deterministic logic programming

Imperative programming

Functional Programing

First Order Functional Programing

Declarative Programing

# Difference Between Structured and Unstructured programming languages:

## Structured Programming Language:

In Structured programming, code is divided into functions or modules. It is also known as modular programming. Each module with set of statements performs a specific task which makes it easy to test and debug. Modification in structured paradigm is also very easy as the programmer only concentrate on a specific module. Examples of Structured programming is C, C++, Java, Python, PERL, PHP etc.

## Unstructured Programming Language:

As explained by the name, Unstructured programming language program is written in a single continuous way. It consists of a single whole block of code. The whole program is taken as a single unit. Doing changes in the code is also very hard. This paradigm also has very limited number of data types This paradigm was used in earlier versions of GWBASIC, COBOL, FROTON etc.

# Difference Between declarative and imperative programing paradigms:

## Declarative Programming:

Declarative Programming express the logic of a computation without describing its control flow. Simply Declarative programming is not like how you do something its like what you do. Declarative code Is hard to understand but logic is always there.

Declarative programming is classified into subtypes. Following are some subtypes of declarative programming.

### Functional Programing:

Functional programming is a programming paradigm where programs are constructed by applying and creating functions or methods. It is a declarative programming paradigm in which function definitions are trees of expressions that each return a value, rather than a sequence. Some Examples of functional programming are Common Lisp, Scheme, Clojure etc.

### Logic Programming:

Logic Programming which is largely based on formal logic. Any program written in a logic programming is a set of sentences in logical form, expressing facts and rules about some problem domain. Examples of Logic programming are Absys, Lisp etc.

### Constraint Programming:

Constraint programming is a paradigm for solving combinative problems. In constraint programming, users declaratively state the constraints on the feasible solutions for a set of decision variables. Examples of constraint programming are Prolog III, CLP and CHIP.

### Dataflow Programming:

Dataflow Programming is a programming paradigm that models a program as a directed graph of the data flowing between operations. It implements dataflow principles and architecture. Examples of Dataflow Programming are ASCET, CAL, Hume etc.

# Imperative Programming:

Imperative Programming express the logic flow in steps. In simple words this programming language is like what to do and how to do. The code of imperative programming is easier to understand then declarative programming language.

Following are the subtypes of Imperative Programming

### Procedural Programming:

Procedural programming is a programming paradigm, based on the concept of the procedure call. Procedures simply contain a series of computational steps to be carried out. Any given procedure might be called at any point during a program's execution. Example of Procedural programming are Fortran, ALGOL, COBOL etc.

### Object-Oriented Programming:

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which can contain data and code: data in the form of fields and code, in the form of procedures or methods. Examples of Object-Oriented Programming are Java, C++, Python, C#, Visual Basic etc.