Object-Oriented Analysis and Design using JAVA (20B12CS334)

B.Tech (CSE/IT) 5th SEM
2021-2022

Brief history of object-oriented software development

Brief history of object-oriented concepts

- The term "object" in programming coined early 1960s, and used in several MIT projects such as Sketchpad (a computer program written by Ivan Sutherland- a pioneer of computer graphics- in 1963 in his PhD thesis-for which he was awarded Turing Award in 1988.

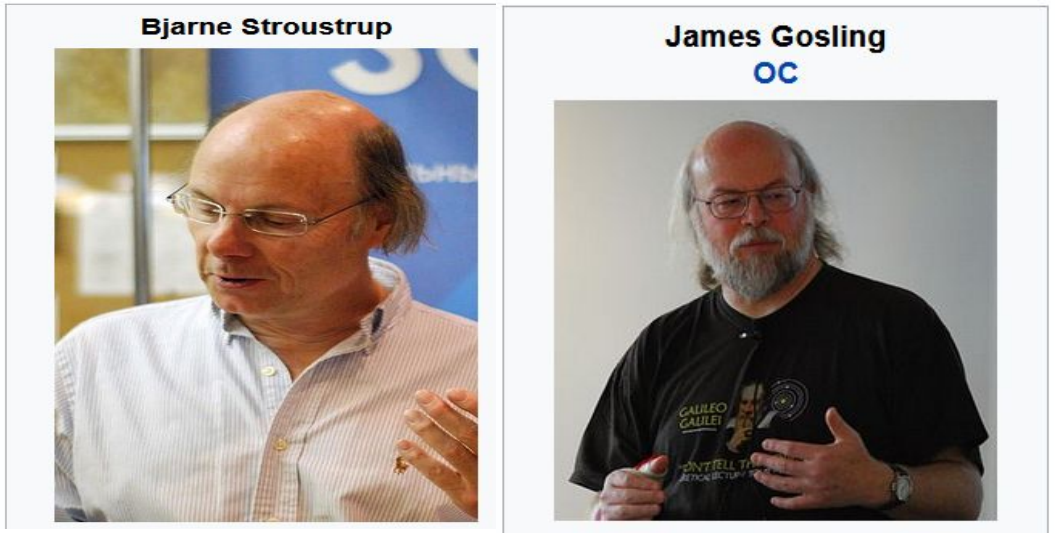
- The Simula-language (1962) is recognized as a first programming language where the core concepts of object-oriented such as classes, objects, inheritance, polymorphism, and dynamic binding introduced.

Ole-Johan Dahl and Kristen Nygaard were a computer scientist and known as father of Simula object-oriented programming. Nygaard and Dahl received the 2001 A. M. Turing Award for their contribution to computer science.

- Simula has been used in a wide range of applications such as simulating very-large-scale integration (VLSI) designs, communication protocols, process modeling, algorithms, and other applications such as computer graphics, typesetting, and education.

- The concepts used in Simula has influenced in development of many other object-oriented programming. The Simula-type objects are reimplemented in C++, Object Pascal, Java, C#, and many other languages.

Computer scientists such as Bjarne Stroustrup, creator of C++ (1985), and James Gosling, creator of Java (1995), have acknowledged Simula as a major influence



- Palo Alto Research Center (PARC) developed *Smalltalk* in the early 1970's.
- *Smalltalk* is considered the first truly object-oriented language.
- 1980 Booch pioneered the concept of object-oriented design (OOD).

TIOBE Index for August 2021

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors

Aug 2021	Aug 2020	Change	Programming Language	Ratings	Change
1	1		C	12.57%	-4.41%
2	3	↑	Python	11.86%	+2.17%
3	2	↓	Java	10.43%	-4.00%
4	4		C++	7.36%	+0.52%
5	5		C#	5.14%	+0.46%
6	6		Visual Basic	4.67%	+0.01%
7	7		JavaScript	2.95%	+0.07%
8	9	↑	PHP	2.19%	-0.05%
9	14	⇈	Assembly language	2.03%	+0.99%
10	10		SQL	1.47%	+0.02%
11	18	⇈	Groovy	1.36%	+0.59%
12	17	⇈	Classic Visual Basic	1.23%	+0.41%
13	42	⇈	Fortran	1.14%	+0.83%
14	8	⇊	R	1.05%	-1.75%
15	15		Ruby	1.01%	-0.03%
16	12	⇊	Swift	0.98%	-0.44%
17	16	↓	MATLAB	0.98%	+0.11%
18	11	⇊	Go	0.90%	-0.52%
19	36	⇈	Prolog	0.80%	+0.41%
20	13	⇊	Perl	0.78%	-0.33%

https://www.tiobe.com/tiobe-index/

Programming Paradigms

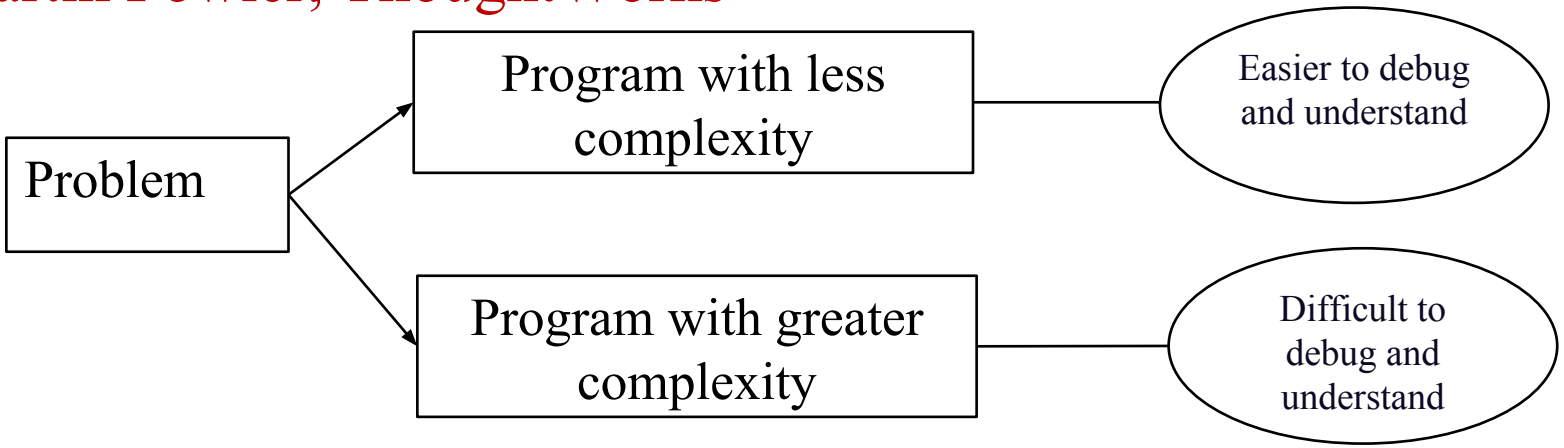
Introduction

- Meaning of the word “paradigm” An example that serves as pattern or model. The American Heritage Dictionary of the English Language.

- Paradigms emerge as the result of social processes in which people develop ideas and create principles and practices that embody those ideas by Thomas Kuhn, “The Structure of Scientific Revolutions”

Programming paradigm

Any fool can write code that a computer can understand. Good programmers write code that humans can understand.
— Martin Fowler, ThoughtWorks

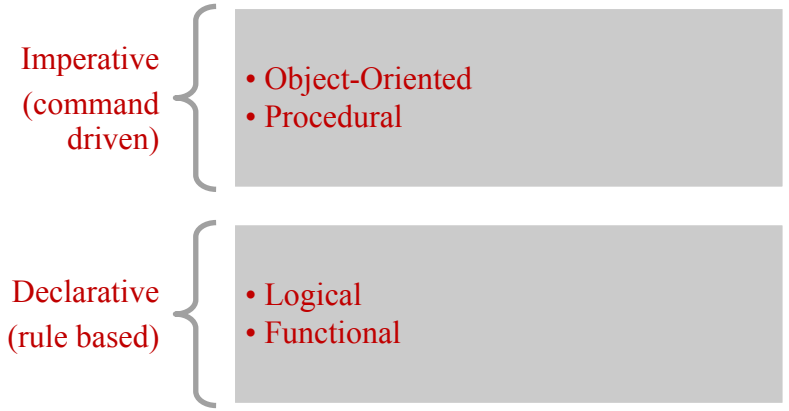


- Managing complexity is a programmer's main concern. So how do programmers deal with complexity? There are many general approaches that reduce complexity in a program or make it more manageable. One of the main approaches is a programming paradigm. Let's dive into programming paradigms!

- The term programming paradigm refers to a style of programming. It does not refer to a specific language, but rather it refers to the way you program.

- There are lots of programming languages that are well-known but all of them need to follow some strategy when they are implemented. And that strategy is a paradigm.

Introduction



- Java
- C++
- Smalltalk
- Eiffel
- Ada++
  - Object oriented
- PROLOG
- GHC
  - Logical
- ML
- Scheme
- Haskell
- Lisp
- CLOS
  - Functional
- C
- COBOL
- FORTRAN
- Ada
- Pascal
  - Procedural

Imperative programming paradigm

The paradigm consists of several statements, and after the execution of all of them, the result is stored. It's about writing a list of instructions to tell the computer what to do step by step.

Object-oriented paradigm

- A collection of classes and object is used.
- Basic entity: object that is used to perform every computation.
- It gives emphasis on data instead procedure.
- All kind of real life problem can be handled.
- Advantage: Reusability, Inheritance, Security, and abstraction.

Procedural paradigm

- This paradigm emphasizes on procedure
- No difference between procedural and imperative approach.
- Advantage to reuse the code.

Declarative programming paradigm

The declarative paradigm focuses on what needs to be done instead of how it should be done. In this, programs are build by expressing logic of computation without defining the control flow.

Logical paradigm

- In this, focus is on knowledge base and the problem.
- The program structure is like mathematical statement

Functional paradigm

- The main concept is the function which define some specific computation.
- Program is structured in such a way that if values are placed instead of function, the logic of the program should not change.
- In this, Data is not much focused. It is loosely coupled to functions

Further programming paradigm

Parallel process paradigm:- Parallel processing is the processing the instructions of program by dividing them among multiple processors.

Objective: Running a program in less time.

The system needs many processor to process the instructions.

Examples: C/C++ as some library function support this paradigm

Database programming paradigm:- This programming focuses on data and its movement.

Program structure defines data instead of series of steps.

Key area is in database programming for business information system.

Functionality: file creation, data entry, update, query and reporting functions

Example: SQL

Literate programming, as a form of imperative programming, structures programs as a human-centered web, as in a hypertext essay: documentation is integral to the program, and the program is structured following the logic of prose exposition, rather than compiler convenience.

Key references

1. Programming Languages: Principles and Paradigms by Maurizio Gabbrielli (Author), Simone Martini (Author).
2. https://www.geeksforgeeks.org/introduction-of-programming-paradigms/
3. https://cs.lmu.edu/~ray/notes/paradigms/#:~:text=A%20programming%20paradigm%20is%20a,thing%20(like%20a%20language).
4. https://en.wikipedia.org/wiki/Programming_paradigm