

20CYS312 - Principles of Programming Languages

Exploring Programming Paradigms

Assignment-01

Presented by Achyuth Anand

CB.EN.U4CYS21002

TIFAC-CORE in Cyber Security

Amrita Vishwa Vidyapeetham, Coimbatore Campus

Feb 2024



AMRITA
VISHWA VIDYAPEETHAM



- 1 Object Oriented Programming
- 2 Object Oriented Programming - Java
- 3 Logic Programming
- 4 Logic Programming - Prolog
- 5 Comparison and Discussions
- 6 Bibliography



- **Introduction to Object-Oriented Programming**

- Object-oriented Programming (OOPs) is defined as the practice of incorporating objects in the program.
- The main aim of OOPs is to bind data and the functions that operate on them to prevent external, unwanted access using ***Class*** and ***Objects***.
- ***Class***: User defined datatype comprising of data members and member functions.
- ***Object***: An instance of a class.



- **Features of OOPs paradigm**

- **Abstraction:** Providing only essential information about the data to the outside world, hiding the background details or implementation.
- **Encapsulation:** Wrapping up of data under a single unit.
- **Inheritance:** The capability of a class to derive properties and characteristics from another class.
- **Polymorphism:** The ability of a message to be displayed in more than one form.
- **Dynamic Binding:** the code associated with a given procedure call is not known until the time of the call at run time
- **Message Passing:** Objects communicate with one another by sending and receiving information to each other.



• Basics of Java

- Java is a programming language that is object-oriented, class-based, and high-level.
- Designed to enable programmers to build code that can be executed on any platform without the need for recompilation.
- Converts the code into **bytecodes** and can run on any Java Virtual Machine(JVM).
- Syntax is heavily influenced by C++ and C.



- OOPs in Java

- **Abstraction**: Done for an entire class using **abstract** keyword
- **Encapsulation**: Instance variable of a class as **private**
- **Inheritance** : Single Inheritance ,Multilevel Inheritance ,Hierarchical Inheritance ,Multiple Inheritance ,Hybrid Inheritance
- **Polymorphism**: Compile-time and run-time Polymorphism.
- **Dynamic Dispatch**: Only method overriding, no overriding of data-members.



- **Common Use cases**

- Designing general software
- Client-Server Systems
- Computer Aided Designs
- Office Automation Systems
- Enhancing security in software



- **Introduction to Logic Programming**

- a computational paradigm that relies on formal logic to program, store and represent knowledge.
- A logic program consists of a collection of phrases expressed in logical form, which serve to represent knowledge pertaining to a specific problem domain.
- Major logic programming language families include Prolog, Answer Set Programming (ASP) and Datalog.



- **Features of Logic Paradigm**

- **Forward Reasoning:** Employs the use of initial facts and data to arrive at the final solution.
- **backward reasoning:** The goal is studied to determine the rules, initial facts, and data.
- **Horn clauses:** logical formula of a particular rule-like form that gives it useful properties for use in logic programming.
- **Negation as Failure:** Attempts to derive $\neg p$, from failure to derive p .
- **Knowledge Representation:** Encoding human knowledge and reasoning (Automated Reasoning) into a symbolic language.



• Basics of Prolog

- Prolog is a programming language for reasoning that comes from the fields of artificial intelligence and computational linguistics.
- prolog comes from firstorder logic, which is a formal logic.
- prolog is mostly meant to be a descriptive programming language, which means that the program is made up of facts and rules that describe relationships.



- Logic in Prolog
 - **Horn clauses**: 2 Kinds, *rules* and *facts*
 - **Negation as a failure**: In pure prolog, NAF literals like $\neg P$ can be found in clause bodies and can be used to get other NAF literals.
 - **Knowledge Representation in Prolog**: predicates stand for knowledge. As in the Relational Model, a predicate is a structure that works like a connection.



- **Common Use cases**

- Widely used in the field of Artificial Intelligence.
- It is used for Natural Language Processing
- The concept of backtracking is an important aspect of machine learning.
- It is used to build expert systems.



- **Purpose**

- ***Object Oriented Programming***: Focuses on development, maintenance and security of software
- ***Logic Programming***: Focuses on problem-solving by representing the problem in terms of logic and facts.

- **Execution method**

- ***Object-oriented Programming***: Compiled in Languages like C++, Java and interpreted in Python and Javascript.
- ***Logic Programming***: Executed using a resolution method, known as backtracking.

- **Error Handling:**

- ***Object Oriented Programming***: Utilises the mechanism of exception handling, error codes, and assertions.
- ***Logic Programming***: Utilises the concept of backtracking to resolve errors, and find new possible solutions



- **Memory Usage**

- ***Object Oriented Programming***: Higher memory utilisation due to Object structures, dynamic memory allocation, and inheritance.
- ***Logic Programming***: Generally associated with lower memory utilisation because of its use of simpler structures, and less run-time overhead for objects and complex data structures.

- **Real world applications**

- ***Object Oriented Programming***: Focuses on software development and maintenance.
- ***Logic Programming***: Focuses on AI, Natural Language Processing(NLP), and expert systems.



- <https://www.geeksforgeeks.org/introduction-of-object-oriented-programming/>
- <https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/>
- https://en.wikipedia.org/wiki/Logic_programming
- https://en.wikipedia.org/wiki/Horn_clause
- https://en.wikipedia.org/wiki/Negation_as_failure
- <https://en.wikipedia.org/wiki/Prolog>
- <https://www.geeksforgeeks.org/prolog-an-introduction/>
- https://en.wikipedia.org/wiki/Knowledge_representation_and_reasoning
- Few Other definitions were taken from the report made, which in turn has cited its own sources.

