# ASSIGNMENT

# OF

# ARTIFICAL INTELLIGENCE

## SUBMITTED TO:- SUBMITTED BY:-

MR.PRIYANK ARORA DIMPLE KUMARI

BCA 6TH (A)

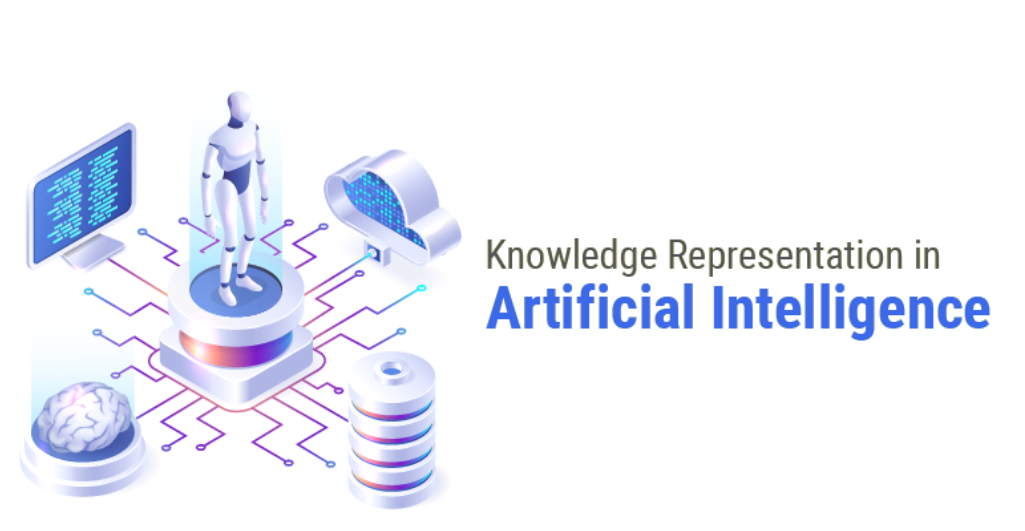
# 19015100

**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATION**

UNIVERSITY INSTITUTE OF COMPUTER APPLICATION AND INFORMATION SCIENCE

## SANT BABA BHAG SINGH UNIVERSITY,JALANDHAR(PUNJAB)

# KNOWLEDGE REPRESENTATION:-



## DEFINITION:-

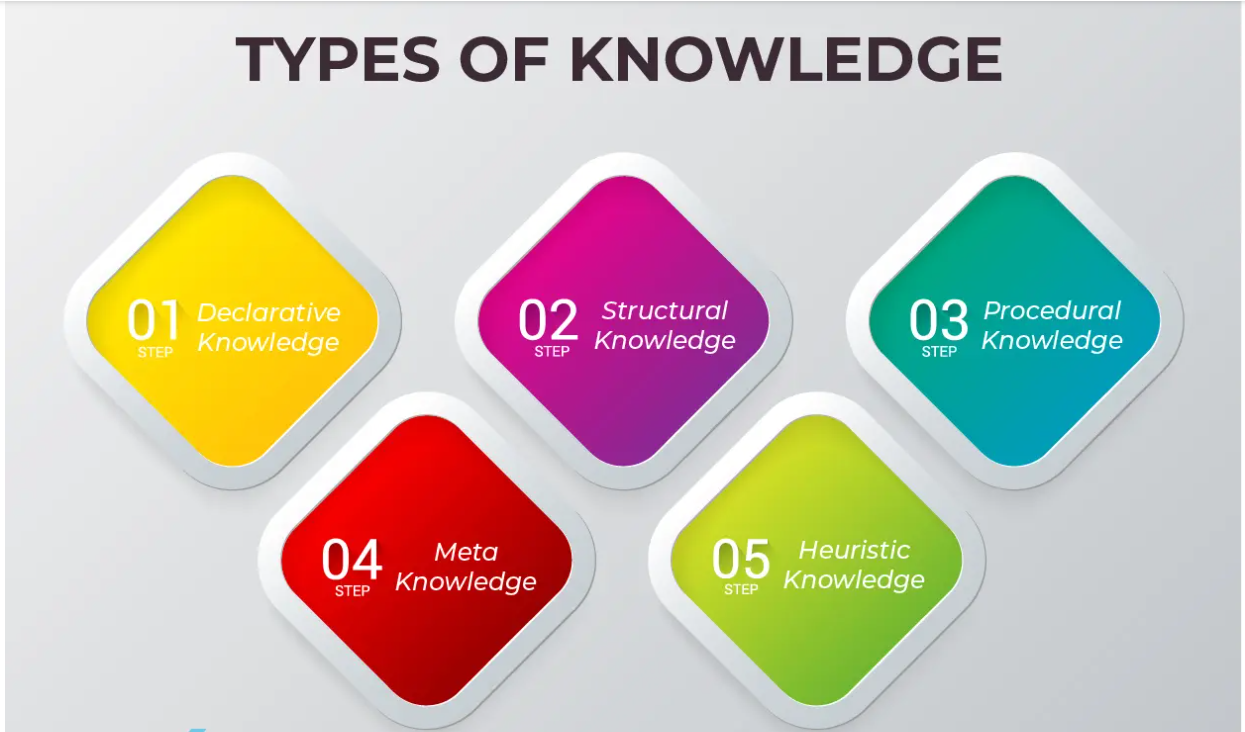
### Knowledge Representation in AI describes the representation of knowledge. Basically, it is a study of how the beliefs, intentions, and judgments of an intelligent agent can be expressed suitably for automated reasoning. One of the primary purposes of Knowledge Representation includes modeling intelligent behavior for an agent.

The different kinds of knowledge that need to be represented in AI include:

* **Objects**
* **Events**
* **Performance**
* **Facts**
* **Meta-Knowledge**
* **Knowledge-base**

| 1 | **OBJECTS** | All the facts about objects in our world domain. E.g., Guitars contains strings, trumpets are brass instruments. |
| --- | --- | --- |
| 2 | **EVENTS** | Events are the actions which occur in our world. |
| 3 | **PERFORMANCE** | It describe behavior which involves knowledge about how to do things. |
| 4 | **FACTS** | Facts are the truths about the real world and what we represent. |
| 5 | **META-KNOWLEDGE** | It is knowledge about what we know. |
| 6 | **KNOWLEDGE-BASE** | The central component of the knowledge-based agents is the knowledge base. It is represented as KB. The Knowledgebase is a group of the Sentences (Here, sentences are used as a technical term and not identical with the English language). |

## TYPES OF KNOWLEDGE REPRESENTATION:-



**1. Declarative Knowledge:-**

* Declarative knowledge is to know about something.
* It includes concepts, facts, and objects.
* It is also called descriptive knowledge and expressed in declarativesentences.
* It is simpler than procedural language.

**2. Structural Knowledge:-**

* Structural knowledge is basic knowledge to problem-solving.
* It describes relationships between various concepts such as kind of, part of, and grouping of something.
* It describes the relationship that exists between concepts or objects.

**3. Procedural Knowledge:-**

* It is also known as imperative knowledge.
* Procedural knowledge is a type of knowledge which is responsible for knowing how to do something.
* It can be directly applied to any task.
* It includes rules, strategies, procedures, agendas, etc.
* Procedural knowledge depends on the task on which it can be applied.

**4. Meta Knowledge:-**

* Knowledge about the other types of knowledge is called Meta-knowledge.

**5. Heuristic Knowledge:-**

* Heuristic knowledge is representing knowledge of some experts in a filed or subject.
* Heuristic knowledge is rules of thumb based on previous experiences, awareness of approaches, and which are good to work but not guaranteed.

## APPROACHES TO KNOWLEDGE REPRESENTATION:-

There are mainly four approaches to knowledge representation, which are givenbelow:

| **S.NO.** | **APPROACHES** | **EXPLANATION** |
| --- | --- | --- |
| 1. | Simple relational knowledge | It is the simplest way of storing facts which uses the relational method, and each fact about a set of the object is set out systematically in columns.  This approach of knowledge representation is famous in database systems where the relationship between different entities is represented.  This approach has little opportunity for inference. |
| 2. | Inheritable knowledge | In the inheritable knowledge approach, all data must be stored into a hierarchy of classes.  All classes should be arranged in a generalized form or a hierarchal manner.  In this approach, we apply inheritance property.  Elements inherit values from other members of a class.  This approach contains inheritable knowledge which shows a relation between instance and class, and it is called instance relation.  Every individual frame can represent the collection of attributes and its value.  In this approach, objects and values are represented in Boxed nodes.  We use Arrows which point from objects to their values. |
| 3. | Inferential knowledge | Inferential knowledge approach represents knowledge in the form of formal logics.  This approach can be used to derive more facts.  It guaranteed correctness. |
| 4. | Procedural knowledge | Procedural knowledge approach uses small programs and codes which describes how to do specific things, and how to proceed.  In this approach, one important rule is used which is **If-Then rule**.  In this knowledge, we can use various coding languages such as **LISP language** and **Prolog language**.  We can easily represent heuristic or domain-specific knowledge using this approach.  But it is not necessary that we can represent all cases in this approach. |

# FIRST ORDER PREDICATE LOGIC:-

* First-order logic is another way of knowledge representation in artificial intelligence. It is an extension to propositional logic.
* FOL is sufficiently expressive to represent the natural language statements in a concise way.
* First-order logic is also known as **Predicate logic or First-order predicate logic**. First-order logic is a powerful language that develops information about the objects in a more easy way and can also express the relationship between those objects.
* First-order logic (like natural language) does not only assume that the world contains facts like propositional logic but also assumes the following things in the world:

**Objects:** A, B, people, numbers, colors, wars, theories, squares, pits, wumpus, ......

**Relations:** **It can be unary relation such as:** red, round, is adjacent, **or n-any relation such as:** the sister of, brother of, has color, comes between

**Function:** Father of, best friend, third inning of, end of, ......

* As a natural language, first-order logic also has two main parts:

**Syntax**

**Semantics**

## Syntax of First-Order logic:-

The syntax of FOL determines which collection of symbols is a logical expression in first-order logic. The basic syntactic elements of first-order logic are symbols. We write statements in short-hand notation in FOL.

### Basic Elements of First-order logic:-

Following are the basic elements of FOL syntax:

38.5M

880

OOPs Concepts in Java

| **Constant** | 1, 2, A, John, Mumbai, cat,.... |
| --- | --- |
| **Variables** | x, y, z, a, b,.... |
| **Predicates** | Brother, Father, >,.... |
| **Function** | sqrt, LeftLegOf, .... |
| **Connectives** | ∧, ∨, ¬, ⇒, ⇔ |
| **Equality** | == |
| **Quantifier** | ∀, ∃ |

### 1. Atomic sentences:-

* Atomic sentences are the most basic sentences of first-order logic. These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms.
* We can represent atomic sentences as **Predicate (term1, term2, ......, term n)**.

**Example: Ravi and Ajay are brothers: => Brothers(Ravi, Ajay).  
        Chinky is a cat: => cat (Chinky)**.

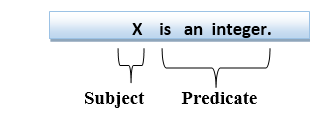
### 2. Complex Sentences:-

* Complex sentences are made by combining atomic sentences using connectives.

**First-order logic statements can be divided into two parts:**

* **Subject:** Subject is the main part of the statement.
* **Predicate:** A predicate can be defined as a relation, which binds two atoms together in a statement.

**Consider the statement: "x is an integer."**, it consists of two parts, the first part x is the subject of the statement and second part "is an integer," is known as a predicate.



## Quantifiers in First-order logic:-

* A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen in the universe of discourse.
* These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression. There are two types of quantifier:

**1.Universal Quantifier, (for all, everyone, everything)**

**2.Existential quantifier, (for some, at least one).**

### Universal Quantifier:-

Universal quantifier is a symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of a particular thing.

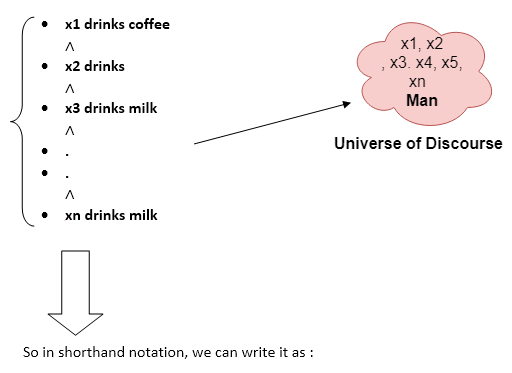
The Universal quantifier is represented by a symbol ∀, which resembles an inverted A.

If x is a variable, then ∀x is read as:

* **For all x**
* **For each x**
* **For every x.**

**EXAMPLE:- All man drink coffee.**

Let a variable x which refers to a cat so all x can be represented in UOD as below:



**∀x man(x) → drink (x, coffee).**

There are all x where x is a man who drink coffee.

## Existential Quantifier:-

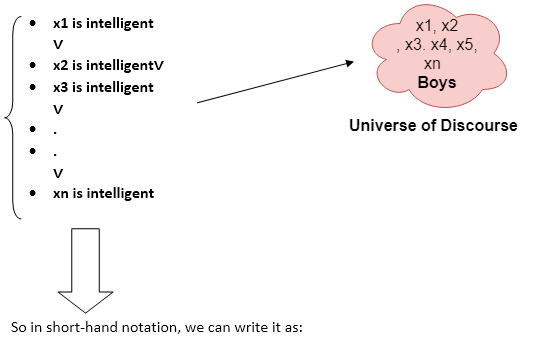
Existential quantifiers are the type of quantifiers, which express that the statement within its scope is true for at least one instance of something.

It is denoted by the logical operator ∃, which resembles as inverted E. When it is used with a predicate variable then it is called as an existential quantifier.

If x is a variable, then existential quantifier will be ∃x or ∃(x). And it will be read as:

* **There exists a 'x.'**
* **For some 'x.'**
* **For at least one 'x.'**

**EXAMPLE :- Some boys are intelligent.**



**∃x: boys(x) ∧ intelligent(x)**

There are some x where x is a boy who is intelligent.