Logic Programming

- Programs in logic languages are expressed in a form of symbolic logic.
- Use a logical inferencing process to produce results.
Logic Programming

- Programs in logic languages are expressed in a form of symbolic logic.
- Use a logical inferencing process to produce results.
- Declarative rather than procedural:
  - Only specification of results are stated (not detailed procedures for producing them)
Symbolic Logic

Logic which can be used for the basic needs of formal logic:

- Express propositions
  - A logical statement that may or may not be true
  - Consists of objects and relationships of objects to each other
Symbolic Logic

Logic which can be used for the basic needs of formal logic:

- Express propositions
  - A logical statement that may or may not be true
  - Consists of objects and relationships of objects to each other
- Express relationships between propositions
- Describe how new propositions can be inferred from other propositions
Atomic propositions consist of compound terms.
Atomic propositions consist of compound terms.

**Compound term**: one element of a mathematical relation, written like a mathematical function:

– Mathematical function is a mapping
Parts of a Compound Term

Compound term composed of two parts

- **Functor**: function symbol that names the relationship.
- Ordered list of **parameters** (tuple).
Parts of a Compound Term

Compound term composed of two parts

- **Functor**: function symbol that names the relationship.
- Ordered list of **parameters** (tuple).
- Examples:
  
  student(jon)
  like(seth, OSX)
  like(nick, windows)
  like(jim, linux)
Forms of a Proposition

- Propositions can be stated in two forms:
  - **Fact**: proposition is assumed to be true
  - **Query**: truth of proposition is to be determined
Propositions can be stated in two forms:
  – **Fact**: proposition is assumed to be true
  – **Query**: truth of proposition is to be determined

**Compound proposition:**
  – Have two or more atomic propositions
  – Propositions are connected by operators
Overview of Logic Programming

- **Declarative semantics**
  - There is a simple way to determine the meaning of each statement
  - Simpler than the semantics of imperative languages
Overview of Logic Programming

- **Declarative semantics**
  - There is a simple way to determine the meaning of each statement
  - Simpler than the semantics of imperative languages

- **Programming is nonprocedural**
  - Programs do not state now a result is to be computed, but rather the form of the result