

# Why Declarative Programming?

- ▶ Properties of imperative (conventional) languages
  - ▶ **State-oriented**: each statement execution changes the abstract machine state
  - ▶ **Destructive assignment** as a fundamental operation
    - ▶ E.g.,  $x = x + 1$
  - ▶ **Side effects** can happen
    - ▶ E.g., modifying a non-local variable
  - ▶ Difficult to read, write, and verify programs

# Why Declarative Programming?

- ▶ Properties of declarative languages
  - ▶ Simple program semantics: “What You See Is What I Mean” (WYSIWIM)
  - ▶ Higher program **understandability** and **verifiability**
  - ▶ **Referential transparency**
    - ▶ Closer to mathematics
    - ▶ Computation by values, not by effects
    - ▶ Everything is deterministic

# Why Declarative Programming?

- ▶ From a software engineering point of view
  - ▶ Correctness is extremely important
  - ▶ The dynamic and interactive environment makes it easy to experiment and change a program while it is being developed
  - ▶ Rapid **prototyping** and **exploratory programming** for problems that are so complex that no clear solution is available at the start of investigation