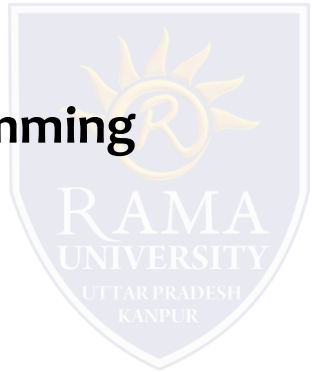
FACULTY OF ENGINEERING & TECHNOLOGY

Brajesh Mishra
Assistant Professor
Department of Computer Science & Engineering

Topics Covered

Functional Programming
Concepts of functional programming
functional programming



Functional Programming

- Functional programming is a programming paradigm in which we try to bind everything in pure mathematical functions style.
- It is a declarative type of programming style.
- Its main focus is on “what to solve” in contrast to an imperative style where the main focus is “how to solve”.
- Programming Languages that support functional programming: Haskell, JavaScript, Scala, Erlang, Lisp, ML, Clojure, OCaml, Common Lisp, Racket.

RAMA
UTTAR PRADESH
KANPUR

Concepts of functional programming

- Pure functions
- Recursion
- Referential transparency
- Functions are First-Class and can be Higher-Order
- Variables are Immutable

RAMA UNIVERSITY
UTTAR PRADESH
KANPUR

Here we briefly look at what functional programming is and why we want to study it

- a function, in the mathematical sense, is a set of operations that perform some computation on the parameter(s) passed, and return a value
- a function, in a programming language, is a set of code whose purpose is to compute based on the parameter(s) and return a value

Functions are often used to promote modularity

- break down program tasks into small, roughly independent pieces
- this promotes structured programming in terms of design
- this also aids debugging, coding and maintenance

However, the function, as we see them in programming languages, does not necessarily reflect a mathematical function

- Functions may act more like procedures
  - a procedure is another unit of modularity
  - the idea behind a procedure is to accomplish one or more related activities
  - the activities should make up some logical goal of the program but may not necessarily be based on producing a single result
  - procedures may return 0 items or multiple items unlike functions
- In languages like C, there are no procedures, so functions must take on multiple roles
  - mathematical types of functions
  - functions that act as procedures
- Such functions can produce side effects
  - mathematical functions do not produce side effects

- Functions should
  - be concise
    - accomplish only a single task or goal
  - return one item
    - in CL, if we need to return multiple values, we can wrap them into a list (or other structure)
      - there is also a way to have CL functions return multiple values although that is generally discouraged
  - have no side effects
    - because the assignment operations are done by function calls, the function calls must produce side effects in such circumstances, but in general, YOUR code should not produce side effects
      - destructive operations are available and are often much more efficient than their non-destructive counterparts, but should not be used haphazardly or just because they are more efficient
  - use parameter passing for communication
    - rather than global variables
  - exploit recursion when possible
    - this simplifies the body of a function and requires fewer or no local variables