PRINCIPLE OF PROGRAMMING LANGUAGES

FALL 2018

Lecture 1

Bahareh Goodarzi

Course Information

- Prof: Bahareh Goodarzi
- Office: EV-3.231
- Email: bahareh.goodarzi@concordia.ca
- Office hours: Thursday: 1 pm-2 pm, or by appointment

Université Concordia University

Tutorials & PODs

- **Tutorials** : **Start Week of September 10.**
- **So NO tutorial this week!**
- **POD Start time will be announced soon.**

- Purpose:
  - Additional exercises and examples
  - Get explanations and help on the assignments

Introduction

- Objectives
- Programming Paradigms
- What do we cover in this course

Principle of Programming Languages

UNIVERSITÉ Concordia UNIVERSITY

Objectives

- To introduce several different paradigms of programming languages
  - *But isn't one language pretty much like another? No!*
- To gain experience with these paradigms by knowing the example programming languages
- To understand concepts of syntax, translation, abstraction, and implementation

UNIVERSITÉ Concordia UNIVERSITY

Programming Language Design

- Language designers have a basic vocabulary about language structure, meaning and pragmatic concerns that help them understand how language works.
- These vocabularies fall into 3 major categories (Programming language design):
  - Syntax
  - Names & Types
  - Semantics

Programming Language Design

- Syntax: Describes what constitutes a structurally correct program
- Names and Types: Enables the programmer to understand and properly implement operations on the values of various types.
- Semantics: Defines the meaning of a program
- i.e. when a program is executed, the effect of each statement on the values of the variable in the program given by the semantic of the language

Programming Paradigms

- A Programming Paradigm is a pattern of problem-solving thoughts that underlines a particular category of programs and languages
- Style of programming refers to
  - Abstraction used to model elements
  - The way computation is being performed
- Different programming paradigms can be best suited to address different types of problems
  - **Programming Paradigms:**
  - Logic Programming
  - Functional Programming
  - Imperative Programming: Includes
    - *Procedural Programming*
    - *Object-oriented Programming*

Programming Paradigms (Cont.)

- **Logic Programming:** describes computation in terms of statements assumed to be logically true. The computation is thus a series of logic deductions. Program statements express facts and rules about problems within a system of formal logic.
  - **Example: Prolog**

- **Functional Programming:** is a programming paradigm that describes a computation in terms of a evaluation of a (stateless) function(which may in turn invoke other functions, etc.). It relies on expressions and declarations rather than statements.
  - **Example: Lisp**

Programming Paradigms (Cont.)

- **Imperative Programming:** Describes computation in terms of statements that change a program state. Types of imperative programming include procedural programming and object oriented programming
  - Procedural Programming: The computation is performed by one or more procedure operating on a collection of data.
    - **Example: C**
  - Object Oriented Programming: computation is performed through a community of inter-communicating agents(objects)
    - **Example : Java**

UNIVERSITÉ Concordia UNIVERSITY

What Do We Cover in This Course?

- Prolog
- Lisp
- Ruby
- C
- ruby
- Aspect-J
- Java