CSE324 Spring14
Principles of Programming Languages

Instructor: Hamdy Soliman **Tel**: 5170
Office: Cramer 211C(inside 214) **Email**: **hss@cs.nmt.edu**
Office Hours: Mon 1:00pm-3:30pm, Wed 1:00pm-2:00pm
(or by appointment, to be scheduled two days ahead, at least)
(Email enquiries are accepted any time)

Text:
" **Principles of Programming Languages**",
B. J. MacLennan. 3ed Edition.

References:
(1) "Programming Languages: Design & Implementation",
Terrence W. Pratt.
(2) "Comparative Programming Languages", Linda W. Friedman.
(3) "Object-Oriented Software Construction", Bertand Meyer.
(4) " Programming Languages: Concepts & Constructs",
Ravi Sethi.

GOALS:

To cover the most distinguished characteristics of several important programming language paradigms. Emphasis will be on concepts and principles of language design, implementation, and use rather than on programming per se. Evaluation of languages from all paradigms (data typing system, syntactic structures, polymorphism, genericity, abstraction, power, security, robustness, ...) will be the center of our discussions. Students should be able to select the best language for programming a given application. Moreover, they should be able to quickly understand and technically critique existing languages, and be ready to design their own.

Grading:

Exam-1	**25%**
Exam-2	**25%**
Homework (paper & programming assignments)	**25%**
Semester Project Report	**15%**
Unannounced quizzes & class participation	**10%**

Class policies:

1) No late assignments (homework/programs) will be accepted!

2) Permission to take exams on other dates than scheduled times will not be granted, except for **extreme** ***medical*** emergencies.

Academic Honesty

Assigned Homeworks and Programming assignments MUST be done in your own, unless it is allowed by the instructor to be a group project. You are required to do homeworks and assignments by yourself. You may not share solutions or drafts of solutions, or even attempts at solutions with other students in the class. Make absolutely sure that you understand NMT's plagiarism policy (in the catalog) and the TCC Usage Policy.

Also, please review http://www.nmt.edu/nmtlib/INFO/ORef/plagiarism.html. If you have any questions, please speak with the instructor.

Important Notices: (IMPORTANT)

- It is **your sole duty** to follow all advertized notices on the class website on daily basis.
- No distraction while lecturing in the class, hence no food/drinks or ringing cell phones.
- **You MUST attend all lectures**, otherwise the instructor is not responsible to provide any material or email you class announcements. Such note applies when you missed a class (or part of the class) **without acceptable justifications and preapproval by the class instructor**.
- **NO makeup** for any assigned work (e.g., exams, homeworks, quizzes, projects, etc) except ONLY in case of extraordinary circumstances that approved by the instructor.
- Late submission of homeworks or programming assignments is subject to point deduction to be decided by the class instructor not less than **15% per every day of being late, up to a max of three days.**

Topics:

--- Introduction and concepts:

* the position of High level languages (HLLs) in the computer system.
* Why HLLs ? * What is a "good" language?
* Compilation of HLLs.

--- HLLs Constructs (Syntax and Semantics)

--- Imperative Languages:
* FORTRAN (static efficiency)
---- Block-Structured Languages:
* ALGOL-60 (Power and Generality)
* PASCAL (Simplicity)
* ADA (Modularity, Portability, and Concurrency)
* MODULA-2 (Modularity, Genericity, and Abstraction)

--- Abstraction and Polymorphism in Some HLLs:
* Basic Definitions * Type Encapsulation (ADT)
* Genericity
* Examples from ADA, Modula-2, and ML.

--- Functional Programming Languages:
Main features of LISP & example progs.
(we might visit ML if time allows)

--- Object Oriented Languages:
* Basic definitions.
* Classes and Inheritance in SMALLTALK, C++, and Java.
* Inheritance and Sub-typing.
* Genericity versus Inheritance.

--- Logic Programming :
Main features of PROLOG and its unification engine.

Topics and grades are subject to change as the course progresses.