

Lenguajes de Programación

Dr. Carlos Artemio Coello Coello
CINVESTAV-IPN
Depto. de Computación
Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco
México, D.F. 07300
e-mail: ccoello@cs.cinvestav.mx
Horario de clases : Martes y Jueves, de 12:00 a 14:00hrs

Enfoque del curso

Estudiaremos los principales temas relacionados con el diseño y la implantación de los lenguajes de programación más representativos de los 4 principales paradigmas existentes en la actualidad: imperativo, funcional, orientado a objetos y lógico. Se estudiará la evolución de las estructuras de datos y de control contenidas en los lenguajes de programación, la motivación para su desarrollo y los compromisos que los diseñadores han tenido que considerar. Veremos cómo la fuerza principal que ha conducido muchas de las decisiones de diseño adoptadas en los lenguajes de programación han sido la búsqueda de una mejor ergonomía y el deseo de incrementar la productividad y confiabilidad en la producción de software. Además, se estudiarán diversos métodos para especificar formalmente la sintaxis de los lenguajes de programación, y los usaremos para ilustrar los compromisos existentes entre facilidad de procesamiento (de una computadora) contra legibilidad (de parte de un humano).

Forma de evaluación

Para calcular la calificación final del curso, se considerarán los siguientes porcentajes:

Tareas	20%
Primer Examen Parcial	15%
Segundo Examen Parcial	15%
Proyecto Final	25%
Examen Final	25%

Para aprobar el curso, deberá obtenerse un mínimo del 70% del total de puntos en juego durante el cuatrimestre (sin incluir los de bonificación).

Temario

- **La era del oscurantismo**
 - Lenguaje máquina
 - Pseudo-código
 - Ensambladores

- **Primera generación: FORTRAN**
 - Historia y motivación
 - Estructuras de datos y control
 - Aportaciones a otros lenguajes
 - Sintaxis
 - Paso de parámetros por referencia
 - Los grandes defectos del lenguaje
 - El debate acerca de su retiro

- **Segunda generación: ALGOL-60**
 - Historia y motivación
 - Estructuras de datos y control
 - Paso de parámetros por nombre
 - Las grandes aportaciones a los lenguajes modernos
 - Herramientas descriptivas: BNF (*Backus-Naur Form*)
 - Aspectos sintácticos
 - Los diagramas de contorno

- **Tercera generación: Pascal**
 - Historia y motivación
 - Estructuras de datos y control
 - Paso de parámetros por valor resultado
 - Los problemas con los arreglos y las cadenas
 - Aspectos sintácticos
 - Apuntadores
 - Estructuras heterogéneas

- Enumeraciones
- **Cuarta generación: Ada**
 - Historia y motivación
 - Estructuras de datos y control
 - Paso de parámetros a la carta
 - Aspectos sintácticos
 - Paquetes
 - Modularidad
 - Concurrencia
 - El mecanismo de comunicaciones *Rendez-Vous*
 - Manejo de excepciones
 - Tareas
 - Sobrecarga de operadores
 - Paquetes genéricos y su comparación con las clases
 - ¿Da Ada las mejores soluciones a todo?
 - Las críticas al mejor lenguaje procedural de la actualidad
- **Programación orientada a objetos: Smalltalk y C++**
 - Historia y motivación
 - Clases, objetos, jerarquías, polimorfismo
 - Envío de mensajes
 - Reglas de ámbito estáticas y dinámicas
 - Las impurezas del C++
 - ¿Es realmente importante la programación orientada a objetos?
- **Programación funcional: LISP, Scheme y ML**
 - Historia y motivación
 - Sintaxis
 - Estructuras de control
 - Recursividad
 - Tipos
 - ¿Por qué importa la programación funcional?
- **Programación orientada a la lógica: PROLOG**
 - Historia y motivación

- Aspectos sintácticos
- Estructuras de control
- Perspectivas y uso en inteligencia artificial
- **Introducción al Cálculo Lambda (opcional)**
 - Igualdad de los términos lambda puros
 - Reglas de sustitución
 - Computación con términos lambda puros
 - Cálculo lambda con tipos
 - Polimorfismo

Libro de Texto

Bruce J. MacLennan, **Principles of Programming Languages: Design, Evaluation, and Implementation**, Third Edition, Oxford University Press, ISBN 0195113063, March 1999.

Textos Complementarios

- D.E. Stevenson, *Programming Language Fundamentals by Example*, Auerbach Publications, Boca Raton, Florida, USA, 2007.
- Maurizio Gabbrielli and Simone Martini, *Programming Languages: Principles and Paradigms*, Springer, London, UK, 2010.
- Ravi Sethi & Tom Stone, *Programming Languages. Concepts and Structures*, Addison-Wesley Publishing Company, Second Edition, 1996.
- Henri E. Bal & Dick Grune, *Programming Language Essentials*, Addison-Wesley Publishing Company, 1994.
- Carlo Ghezzi & Mehdi Jazayeri, *Programming Language Concepts*, John Wiley & Sons, Third Edition, 1997.
- David Gelernter & Suresh Janagannathan, *Programming Linguistics*, MIT Press, 1980.
- George Springer & Daniel P. Friedman, *Scheme and the Art of Programming*, The MIT Press, 1990.
- Harold Abelson & Gerald Jay Sussman, *Structure and Interpretation of Computer Programs*, The MIT Press, 1985.

Material para el curso

Un conjunto de lecturas adicionales para el curso se encuentran disponibles en:
<http://delta.cs.cinvestav.mx/~ccoello/lenguajes/bibleng.html>

La página del curso se encuentra en:

<http://delta.cs.cinvestav.mx/~ccoello/lenguajes/>

En esta página se irán colocando también los acetatos utilizados por el instructor en clase, así como las tareas asignadas a lo largo del cuatrimestre.

Una sección de cada tarea, incluirá problemas en Scheme. El aprendizaje de este lenguaje lo deberá realizar de manera auto-didacta, y podrá apoyarse en las notas proporcionadas por el instructor en la página del curso. En estas notas se utiliza el **Scheme 48**, el cual puede descargarse (hay versiones para Windows y para Linux) de:

<http://s48.org/>

Avisos importantes

Todas las tareas entregadas después de la fecha acordada, serán PENALIZADAS con un 10% MENOS SOBRE LA CALIFICACIÓN OBTENIDA POR CADA PERIODO DE 24 HORAS POSTERIOR A LA HORA DE ENTREGA, a menos que otras penalizaciones se estipulen específicamente para una cierta tarea (por ejemplo, que no se admitirá si se entrega después de la fecha indicada). La hora de entrega de todas las tareas será al INICIO de la clase, a menos que se estipule lo contrario. Tampoco se admitirán tareas por partes (es inadmisibles, por ejemplo, querer entregar un día tarde sólo uno de los problemas de una tarea).

Una vez que una tarea haya alcanzado un valor de cero, la penalización se detendrá (es decir, no se considerarán calificaciones negativas), sin embargo, LA ENTREGA DE TODAS LAS TAREAS Y DEL PROYECTO FINAL ES OBLIGATORIA PARA PODER APROBAR EL CURSO, aunque el puntaje asignado a cualquiera de ellas (incluyendo el proyecto final) no afectara de manera significativa la calificación final de un alumno.

Las tareas y los exámenes contendrán preguntas opcionales de bonificación que, de ser contestadas correctamente, permitirán recuperar puntos perdidos a lo largo del cuatrimestre. Esto implica que será posible obtener más del 100% de los puntos en una tarea o examen, de contestarse todas las preguntas correctamente, incluyendo las de bonificación. **Las tareas entregadas tardíamente NO recibirán puntos de bonificación.**

El examen final, sin embargo, será obligatorio para todos, independientemente del puntaje alcanzado a lo largo del cuatrimestre.

El instructor considera como una ofensa muy seria el plagio de tareas de cualquier fuente bibliográfica y/o de otros compañeros, y en caso de comprobarse éste, se REPROBARÁ DE FORMA AUTOMÁTICA A TODOS LOS INVOLUCRADOS.

Es necesario que todos los alumnos proporcionen una dirección electrónica en la lista que se circulará en la clase, y que LEAN FRECUENTEMENTE SU CORREO ELECTRÓNICO, pues muchas de las dudas y avisos importantes se transmitirán por este medio. De tal forma, es total responsabilidad de los alumnos enterarse oportunamente de lo que se transmita por correo electrónico.