

# Conceptos de Lenguajes de Programación

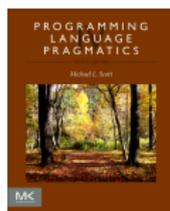
Alberto Pardo    Marcos Viera

Instituto de Computación, Facultad de Ingeniería  
Universidad de la República, Uruguay

El estudio de los **Lenguajes de Programación**, tanto desde el punto de vista de los **conceptos fundamentales** que son comunes a los lenguajes como en lo que respecta a sus diferentes **opciones de diseño**.

- Seminario

- Cada semana estudiaremos un capítulo del libro:



Michael L. Scott

**Programming Language Pragmatics**

Morgan Kaufmann (2015). 4th Edition.

- Un grupo presenta el capítulo
- Todos leen y participan en la discusión
- Ejercicios individuales entregables

- Evaluación

- Asistencia mínima del 60% de las clases (8 de 13)
- Nota de aprobación:
  - 30% presentación
  - 50% prácticos individuales entregables
  - 20% participación en clase.

# Elección del Cupo

- Período de inscripción:  
22/07/24 al 04/08/24
- Generación de información por parte de bedelía:  
05/08/24 al 08/08/24
- Sorteo del cupo y publicación de lista ordenada:  
12/08/24 (o antes, si es posible)
- Período formal de aceptación del cupo:  
13/08/24 al 22/08/24.
- Publicación de estudiantes asignados:  
23/03/24

# Cronograma Tentativo

06/08	Introducción al Curso Introducción a los Lenguajes de Programación
13/08	Libre
20/08	Sintaxis: Gramáticas Libres de Contexto, Scanning, Parsing
27/08	Nombres, Alcance y Ligaduras: Tiempo de vida de un objeto, Reglas de Alcance, Ambientes de Ejecución
03/09	Análisis Semántico: Gramáticas de Atributos
10/09	Control de Flujo: Secuencia, Selección, Iteración, Recursión, No Determinismo
17/09	Sistemas de Tipos: Chequeo de Tipos, Polimorfismo
24/09	Parciales
01/10	Parciales
08/10	Tipos Estructurados: Registros, Arrays, Strings, Conjuntos, Punteros, Listas
15/10	Subrutinas: Pasaje de Parámetros, Manejo de Excepciones, Corutinas, Eventos
22/10	Abstracción de Datos y Orientación a Objetos
29/10	Programación Funcional
05/11	Programación Lógica
12/11	Concurrencia
19/11	Lenguajes de Scripting

# Evolución de los Lenguajes de Programación

## Lenguaje de Máquina

```
55 89 e5 53 83 ec 04 83 e4 f0 e8 31 00 00 00 89 c3 e8 2a 00
00 00 39 c3 74 10 8d b6 00 00 00 00 39 c3 7e 13 29 c3 39 c3
75 f6 89 1c 24 e8 6e 00 00 00 8b 5d fc c9 c3 29 d8 eb eb 90
```

```
    pushl %ebp                                jle    D
    movl  %esp, %ebp                          subl  %eax, %ebx
    pushl %ebx                                B:    cmpl  %eax, %ebx
    subl  $4, %esp                             jne   A
    andl  $-16, %esp                          C:    movl  %ebx, (%esp)
    call  getint                               call  putint
    movl  %eax, %ebx                          movl  -4(%ebp), %ebx
    call  getint                               leave
    cmpl  %eax, %ebx                          ret
    je    C                                   D:    subl  %ebx, %eax
A:    cmpl %eax, %ebx                          jmp   B
```

## Assembler

```
int gcd(int a, int b) {                               // C
    while (a != b) {
        if (a > b) a = a - b;
        else b = b - a;
    }
    return a;
}

let rec gcd a b =                                     (* OCaml *)
    if a = b then a
    else if a > b then gcd b (a - b)
    else gcd a (b - a)

gcd(A,B,G) :- A = B, G = A.                          % Prolog
gcd(A,B,G) :- A > B, C is A-B, gcd(C,B,G).
gcd(A,B,G) :- B > A, C is B-A, gcd(C,A,G).
```

## Lenguajes de Alto Nivel