Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area V: Automatas y Lenguajes

(Programa del año 2018)
(Programa en trámite de aprobación)
(Presentado el 21/11/2018 10:41:59)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
ANALISIS COMPARATIVO DE LENGUAJES	PROF.CS.COMPUT.	007/05	2018	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
ROGGERO, PATRICIA BEATRIZ	Prof. Responsable	P.Adj Exc	40 Hs
ERRECALDE, MARCELO LUIS	Prof. Colaborador	P.Asoc Exc	40 Hs
FUNEZ, DARIO GUSTAVO	Responsable de Práctico	JTP Exc	40 Hs
GATICA, CLAUDIA RUTH	Auxiliar de Práctico	A.1ra Semi	20 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
Hs	3 Hs	4 Hs	1 Hs	8 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
06/08/2018	17/11/2018	15	120

IV - Fundamentación

Este curso trata sobre los aspectos principales del diseño e implementación de los lenguajes de programación. También se incluyen la teoría y los modelos formales subyacentes que forman la base para las decisiones tomadas en la creación de este tipo de lenguajes.

En este sentido, el curso complementa los contenidos introducidos en Programación I y Programación II, donde el énfasis fue puesto en los aspectos de programación utilizando lenguajes representativos de algunos de los principales paradigmas de programación.

En este curso la descripción de los lenguajes se realiza por medio de la arquitectura de hardware y software subyacente que se requiere para la ejecución de programas escritos en estos lenguajes.

Los modelos formales que dan origen a los principales paradigmas de programación también son analizados con una visión más general.

Los contenidos del curso se corresponden con las unidades de conocimiento recomendadas por la ACM/IEEE Computer Society Joint Curriculum Task Force, para el área de lenguajes de programación.

En el convencimiento de que los aspectos básicos del diseño de compiladores deberían formar parte del conocimiento general de cualquier buen programador, este curso muestra de qué manera son compiladas las distintas construcciones de los lenguajes.

Los aspectos abordados, relacionados a la teoría formal de lenguajes servirán además como introducción a tópicos más avanzados a desarrollarse en otros cursos de la carrera como Autómatas y Lenguajes, y Compiladores.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso los alumnos serán capaces de:

Evaluar en forma crítica distintos lenguajes de programación existentes y futuros.

Responder a cuestiones tales como las motivaciones de la existencia de tantos lenguajes de programación, cómo y porqué fueron desarrollados, en qué se asemejan y difieren.

Realizar un curso sobre el estudio del diseño y construcción de compiladores.

Reconocer los paradigmas claves usados en el desarrollo de lenguajes de programación modernos, sus bases teóricas, aplicativas y de implementación.

Entender la implementación de distintos lenguajes con suficiente detalle como para reconocer la relación entre un programa fuente y su comportamiento en ejecución.

Extender sus conocimientos sobre los tópicos anteriores con bibliografía adecuada y mínima supervisión.

VI - Contenidos

UNIDAD 1

Razones del estudio de lenguajes de programación. Historia de los lenguajes de programación. Características de un buen lenguaje. La estructura y operación de una computadora. Computadora de hardware, de firmware. Traductores e Intérpretes. Computadoras virtuales y ligaduras. Tiempos de ligadura. Paradigmas de lenguajes.

UNIDAD 2

Programación Lógica. El lenguaje Prolog. Elementos básicos: hechos, predicados y consultas. Átomos y variables. Variables anónimas. Matching. Reglas recursivas. Proceso de backtracking. Functores. Listas.

UNIDAD 3

Sintaxis de los lenguajes de programación. Criterios sintácticos generales. Nociones básicas de semántica formal. Elementos sintácticos de un lenguaje. Estructura programa-subprograma general. Etapas en la traducción. Modelos de traducción formales. Gramáticas Libres del Contexto. Ambigüedad. Gramáticas Regulares. Autómatas finitos. Expresiones regulares. Autómatas Pushdown.

UNIDAD 4

Propiedades de tipos y objetos. Tipos de datos. Especificación e implementación de tipos de datos elementales. Operaciones. Declaraciones. Chequeo de tipos. Conversión de tipos. Asignación e inicialización. Tipos de datos numéricos. Enumeraciones. Booleanos. Caracteres. Tipos de datos estructurados. Especificación e implementación. Vectores y arreglos. Registros. Cadenas de caracteres. Punteros. Conjuntos. Equivalencia de Tipos.

UNIDAD 5

Definición y activación de subprogramas. Administración de memoria. Fases de la administración de memoria. Administración de memoria estática. Administración de memoria basada en Pila. Heap con elementos de tamaño fijo y de tamaño variable. Mecanismos de Recuperación.

UNIDAD 6
Control de subprogramas. Llamada-retorno simple. La regla de la copia. Subprogramas recursivos. Control de datos. Ambientes de referenciación. Alcance estático y dinámico. Estructura de bloques. Datos compartidos en subprogramas. Parámetros. Pasaje de parámetros. Ambientes comunes explícitos. Alcance dinámico, reglas e implementaciones. Alcance estático, reglas e implementaciones.

UNIDAD 7
Variantes en control de subprogramas. Excepciones y manejadores de excepciones. Corutinas. Subprogramas planificados. Programación paralela: comandos en guardia, tareas.

UNIDAD 8
Control de secuencia explícito e implícito. Secuenciamiento en expresiones aritméticas. Representación de árbol. Representación en tiempo de ejecución. Control de secuencia entre sentencias. Sentencias estructuradas.

UNIDAD 9
Tipos de datos abstractos. Evolución del concepto de tipo de datos. Ocultamiento de la información. Encapsulamiento mediante subprogramas. Tipos de datos abstractos genéricos. Programación orientada a objetos: herencia, polimorfismo y ligadura dinámica. Subclases y subtipos. Herencia simple y múltiple. Aspectos de diseño en Smalltalk, C++ y Java.

VII - Plan de Trabajos Prácticos

Los prácticos de la asignatura son del tipo:
a) Prácticos de aula, con un total de 60 horas.
b) Prácticos de formación experimental, con un total de 15 horas.

Práctico 1: Computadora Virtual, Ligadura.
Objetivos: visualizar las características de un buen lenguaje, ejercitar sobre los distintos tipos de ligaduras.
Metodología: resolución de ejercicios en lápiz y papel.

Práctico 2: Prolog (aula y formación experimental).
Objetivos: comprender las características de un lenguaje lógico, aprender los aspectos claves del lenguaje Prolog, reglas de matching, recursión y estructuras.
Metodología: resolución de ejercicios en lápiz y papel. Probar dichos ejercicios en máquina.

Práctico 3: Descripción Formal de Lenguajes.
Objetivos: ejercitar con autómatas, gramáticas y expresiones regulares.
Metodología: resolución de ejercicios en lápiz y papel.

Práctico 4: Tipos de Datos Elementales y Estructurados.
Objetivos: ejercitar sobre operaciones, signatura, declaraciones, tipos de datos elementales y estructurados. Chequeo de tipos.
Metodología: resolución de ejercicios en lápiz y papel.

Práctico 5: Administración de Memoria (aula y formación experimental).
Objetivos: ejercitar definición y activación de subprogramas, mecanismos de administración de memoria, problemas asociados al uso de punteros y métodos de recuperación.
Metodología: resolución de ejercicios en lápiz y papel.

Práctico 6: Control de Datos en Subprogramas.
Objetivo: realizar ejercicios referidos a ambientes de referenciación, implementación de reglas de alcance estático y dinámico.
Metodología: resolución de ejercicios en lápiz y papel.

Práctico 7: Variantes en control de subprogramas.
Objetivo: desarrollo de ejercicios con variantes en control de subprogramas: excepciones, corutinas, subprogramas planificados.
Metodología: resolución de ejercicios en lápiz y papel.

Práctico 8: Programación Orientada a Objetos.
Objetivos: ejercitar sobre distintos aspectos de la programación orientada a objetos (ligadura dinámica, herencia, polimorfismo).
Metodología: resolución de ejercicios en lápiz y papel.

Prácticos de Formación Experimental:

Laboratorio: Lenguaje Prolog. Realización de un proyecto, en etapas, utilizando el lenguaje.
Objetivos: evaluar consultas, aplicación de las reglas de matching, programación del proyecto.
Metodología: resolución en computadora de escritorio.

VIII - Regimen de Aprobación

A. Régimen para alumnos Regulares:

1) Aprobar el práctico de laboratorio solicitado.

2) Entregar resueltos, al menos el 80% de los ejercicios de prácticos de aula que la cátedra pudiera solicitar.

3) Aprobar 1 examen parcial, de los temas prácticos vistos en la materia, con al menos el 70% correcto del total.

Tal como lo establece la reglamentación vigente, Ord. 32/14 CS, el examen parcial tendrá dos (2) recuperaciones.

4) Se requiere el 70% de asistencia a clases teóricas y prácticas.

B. La materia no admite régimen promocional.

C. Para aprobar la materia el alumno deberá rendir un examen final, el cual podrá ser oral y/o escrito.

D. No se admite rendir la materia en condición de libre.

IX - Bibliografía Básica

[1] "Programming Languages - Design and Implementation". Pratt, Terrence y Zelkowitz, Marvin. Cuarta edición. Prentice Hall, 2001.
[2] "Concepts of Programming Languages". Sebesta, Robert. Addison-Wesley. Edición 2016 y Edición 2004.
[3] "Prolog, programming for artificial intelligence". Bratko, Ivan. Addison-Wesley. Tercera Edición, 2001.
[4] "El lenguaje de Programación C". Kernighan, Brian y Ritchie, Dennies. Prentice Hall, 1991.
[5] Apunte de la Cátedra Aspectos formales de la traducción de lenguajes
[6] Apunte de la Cátedra de Evolución del Concepto de Tipos de Datos - Tipo de Dato Abstracto - Programación Orientada a Objetos.
[7] Apunte de la Cátedra del Lenguaje Prolog.

X - Bibliografia Complementaria

[1] "Lenguajes de Programación - Diseño e Implementación". Pratt, Terrence y Zelkowitz, Marvin. Tercera edición. Prentice Hall, 1999.
[2] "Smalltalk-80. The Language and its implementation". Goldberg, Adele y Robson, David. Addison-Wesley, 1985.
[3] The Java class libraries. Chan, Patrick - Lee, Rosanna y Kramer, Douglas. Addsion Wesley. Segunda Edición, 1998.

[4] "Piensa en Java". Eckel Bruce. Pearson Alhambra. Cuarta Edición, 2007.
[5] "Introduction to Java". Carlos Kavka. ICTP, 2004.
[6] "Introduction to Java". Hume, J.N. Patterson y Stephenson, Christine. Canada Holt Software Associates. Primera Edición, 2000.

XI - Resumen de Objetivos

El curso tiene como objetivo introducir al alumno a la problemática del diseño e implementación de lenguajes, incluyendo fundamentos teóricos y modelos formales. El estudio se realiza teniendo en cuenta todos los paradigmas actuales de programación, realizando un estudio comparativo de las técnicas de implementación de cada uno de ellos.

XII - Resumen del Programa

Historia de los lenguajes de programación. Evolución de los paradigmas de programación. Computadoras virtuales. Sistemas de traducción. Sintaxis y semántica. Gramáticas, expresiones regulares, autómatas. Características esenciales de los lenguajes de programación y su implementación: tipos de datos y su representación, control de secuencia y datos. Administración de Memoria. Abstracción de Datos. Programación lógica, lenguaje Prolog. Control de datos a nivel de subprogramas. Variantes en el control de subprogramas.

XIII - Imprevistos

XIV - Otros

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	