Ministerio de Cultura y Educación **(Programa del año 2015)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
PROGRAMACION II	LIC.CS.COMP.	32/12	2015	1° cuatrimestre
PROGRAMACION II	PROF.CS.COMPUT.	02/16	2015	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
NECCO, CLAUDIA MONICA	Prof. Responsable	P.Adj Exc	40 Hs
ALBORNOZ, MARIA CLAUDIA	Auxiliar de Práctico	A.1ra Semi	20 Hs
BERNARDIS, EDGARDO	Auxiliar de Práctico	A.1ra Simp	10 Hs
ESCUDERO, LEONARDO RUBEN	Auxiliar de Práctico	A.2da Simp	10 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
3 Hs	0 Hs	3 Hs	3 Hs	9 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
16/03/2015	26/06/2015	15	135

IV - Fundamentación

En este curso se inicia a los alumnos en dos paradigmas: prog. Orientada a Objetos y prog. Declarativa Funcional y se refuerzan y completan los conocimientos y aptitudes que adquirieron en Programación I en el paradigma de la programación estructurada. El contenido mencionado prepara al alumno para poder comparar las características de los distintos modelos formales subyacentes en el desarrollo de programas en los tres paradigmas. Las aptitudes desarrolladas a través de la práctica en esta materia, servirán de base tanto para el desarrollo de aplicaciones reales en los distintos paradigmas como para el análisis comparativo formal lenguajes.

También se explora el tema de verificación de programas continuando con el estilo de demostración semántico funcional de Dijkstra (introducido en materia anterior), usando alguna herramienta de software existente para realizar aserciones sobre programas simples, reforzando las bases para su posterior tratamiento formal tanto en el ámbito de la ingeniería del software como en el de teoría de la computación.

V - Objetivos / Resultados de Aprendizaje

- Iniciar el estudio del paradigma de la programación funcional
- Ejercitar desarrollos de programas basados en el paradigma de la prog. funcional
- Iniciar el estudio y aplicación del paradigma de la prog. orientada a objetos
- Ejercitar desarrollos de programas basados en el paradigma de la prog. orientada a objetos

-Continuar el estudio y aplicación de técnicas de verificación de software, en particular del estilo de demostración semántico functional de Dijkstra usando alguna herramienta para verificar software previamente desarrollado.

VI - Contenidos

1.Programación Funcional

Características generales de los lenguajes funcionales. Estilo de programación pointfree vs poitwise. Evaluación perezosa vs Evaluación impaciente. Transparencia referencial. Concepto de función, definición y aplicación. Pattern matching. Funciones recursivas. Funciones de orden superior, definición y aplicación. Currificación y aplicación parcial de funciones. Composición de funciones. Manejo de listas. Listas por comprensión.

Lenguaje asociado: Haskell.

Entorno de trabajo, definición de programas, uso del intérprete.

Módulos. Notación de listas [n..m]. Notación de listas por comprensión. Operadores infijos y prefijos. Reglas de precedencia. Funciones predefinidas para manejo de listas y tuplas. Funciones predefinidas de orden superior. Análisis de funciones recursivas. Recursión de cola, uso de acumuladores. Análisis y uso de las funciones de folding para listas predefinidas: foldl, foldl1, foldr, foldr1.

1. Programación Orientada a Objetos

Descripción del paradigma de orientación a Objetos. Objetos. Mensajes. Clases.

Conceptos: super, self. Encapsulamiento. Herencia. Control de herencia. Herencia multiple. Sobrecarga y Polimorfismo.

Lenguaje asociado:Java

Características generales. Tipos de datos básicos. Estructuras de control. Clases, objetos, variables y métodos. Declaración , creación y destrucción de Objetos. Control de acceso. Herencia. Gestión de excepciones en java. Programación de interfaces gráficas.

3.Verificación de Programas

Descripción y características generales de técnicas de verificación basadas en: casos de prueba, model-checking y en la utilización de métodos formales. Metodos formales: Introducción al estilo semántico functional de Dijkstra. Transformadores de programas. Precondición más débil (weakest precondition WP). Determinismo y disyuntividad. Iteraciones. Aserciones, Invariantes.

Lenguaje asociado: ESC/Java2.

VII - Plan de Trabajos Prácticos

Prácticos de aula:

1. Desarrollo de ejercicios en Lenguaje Java
2. Desarrollo de ejercicios en Lenguaje Haskell
3. Desarrollo de ejercicios de demostración de algorítmos, sobre implementaciones concretas de problemas tipo.

Prácticos de máquina:

Desarrollo y entrega de prácticos de los temas vistos, entre 1 y tres en total según criterio de la cátedra.

VIII - Regimen de Aprobación

- Para regularizar la asignatura:

el alumno debe aprobar dos exámenes parciales o sus correspondientes recuperaciones, y presentar y aprobar en forma y tiempo los prácticos de máquina solicitados por la cátedra. Para aprobar la materia deberán rendir un examen final.

- Para promocionar la asignatura:

el alumno debe cumplir con las condiciones de regularización y aprobar los exámenes parciales con un nivel superior o igual al 70% del total y

resolver los ejercicios señalados particularmente en cada examen parcial (evaluación integradora) con un nivel superior o igual al 70% del total.

Se tomará una recuperación general adicional a los alumnos que hayan presentado certificado de trabajo en tiempo y forma.

EXAMEN LIBRE:

Consta de dos partes:

1) Desarrollo y entrega de prácticos de máquina.

2) Exámen teórico-práctico.

Los alumnos que entreguen y aprueben los prácticos de máquina (en tiempo y forma) podrán presentarse al exámen teórico-práctico.

IX - Bibliografía Básica

[1] Object-Oriented Programming: An Evolutionary Approach, Addison-Wesley Pub (Sd); 2 Sub edition (May 1991), ISBN-10: 0201548348, ISBN-13: 978-0201548341.
[2] Thinking in Java (4th Edition),Bruce Eckel, Prentice Hall; 4 edition (February 20, 2006), ISBN-10: 0131872486, ISBN-13: 978-0131872486.
[3] Sun Microsystems, The Java tutorial. A practical guide for programmers. Available online at http://java.sun.com
[4] Haskell: The Craft of Functional Programming (2nd Edition), Simon Thompson, April 8, 1999, Addison-Wesley, ISBN 0-201- 34275-8.
[5] A Discipline of Programming, Edsger W. Dijkstra, Prentice Hall, Inc. (October 28, 1976), ISBN-10: 013215871X, ISBN-13: 978-0132158718
[6] The Science of Programming (Monographs in Computer Science), David Gries, Springer (February 1, 1987), ISBN-10: 0387964800, ISBN-13: 978-0387964805

X - Bibliografia Complementaria

[1] Flanagan D, Java in a Nutshell: A desktop quick reference (3rd Edition), O'Reilly, 1999.
[2] Principles of Functional Programming. Hugh Glaser, Chris Hankin, David Till.
[3] Predicate Calculus and Program Semantics (Monographs in Computer Science), Edsger W. Dijkstra, Carel S. Scholten Springer; 1 edition (December 18, 1989), ISBN-10: 0387969578, ISBN-13: 978-0387969572

XI - Resumen de Objetivos

1. El paradigma de la Programación Orientada a Objetos:
Descripción del paradigma de orientación a Objetos. Lenguaje Java
2. El paradigma de la Programación Funcional:
Características generales de los lenguajes funcionales. Lenguaje de programación Funcional Haskell
3. Verificación de Programas:
Práctica del estilo de demostración semántico funcional de Dijkstra usando alguna herramienta particular para probar propiedades de algoritmos simples codificados en alguno de los lenguajes conocidos por el alumno.

XII - Resumen del Programa

1. El paradigma de la Programación Orientada a Objetos:
Descripción del paradigma de orientación a Objetos. Lenguaje Java
2. El paradigma de la Programación Funcional:
Características generales de los lenguajes funcionales. Lenguaje de programación Funcional Haskell
3. Verificación de Programas:
Consolidación de las bases del estilo de demostración semántico funcional de Dijkstra, mediante el uso de herramienta concreta de software.

XIII - Imprevistos

XIV - Otros