Ministerio de Cultura y Educación **(Programa del año 2020)**
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.	18/11	2020	1° cuatrimestre
PROGRAMACION II	LIC.CS.COMP.	006/05	2020	1° cuatrimestre
PROGRAMACION II	PROF.CS.COMPUT.	06/09	2020	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
NECCO, CLAUDIA MONICA	Prof. Responsable	P.Asoc Exc	40 Hs
ALBORNOZ, MARIA CLAUDIA	Responsable de Práctico	JTP Semi	20 Hs
MAGUIRE, MARGARITA	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	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**
09/03/2020	19/06/2020	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, basado el la lógica axiomática de Hoare (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, basado en la lógica axiomática de Hoare, usando alguna herramienta para verificar software previamente desarrollado.

VI - Contenidos

1.Programación Funcional

Lenguaje a utilizar: Haskell.

Conceptos Generales del paradigma:

Características de los lenguajes funcionales. Estilo de programación libre de puntos (pointfree) vs atada a puntos (poitwise). Evaluación perezosa vs Evaluación impaciente. Transparencia referencial. Concepto de función, definición y aplicación. Coincidencia de patrones (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.

Conceptos particulares del paradigma en el lenguaje Haskell:

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

Módulos. Notación de listas por comprensión. Operadores infijos y prefijos. Reglas de precedencia.

Funciones predefinidas para manejo de listas y tuplas. Definición y uso de Funciones de orden superior. Expresiones Lambda. Definición y manipulación de funciones y tipos polimorficos. Análisis de funciones recursivas. Recursión de cola vs recursión controlada (tail vs guard recursion), uso de acumuladores. Análisis y uso de las funciones de folding predefinidas para listas: foldl, foldl1, foldr, foldr1.

2. Programación Orientada a Objetos

Lenguaje a utilizar: Java

Conceptos Generales del paradigma:

Descripción del paradigma de orientación a Objetos. Conceptos: clases, objetos, servicios/métodos, estado/atributos, mensajes, super, self, encapsulamiento. Herencia. Control de herencia. Tipos de herencia. Sobrecarga y Polimorfismo.

Conceptos particulares del paradigma en el lenguaje Java:

Características generales del lenguaje. Tipos de datos básicos. Estructuras de control. Clases, objetos, variables y métodos. Declaración , creación y destrucción de Objetos. Control de acceso. Herencia. Polimorfismo. Compatibilidad de objetos. Conversión de tipos (Down/Up Casting). Enlace estático vs dinámico de métodos. Definición e implementación de Interfaces. Gestión de excepciones en java. Programación de Interfaces de Usuario Textuales (IUT/) y Gráficas (IGU).

3.Verificación de Programas

Herramienta a utilizar: Key-Hoare Tool

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

VII - Plan de Trabajos Prácticos

Prácticos de aula:

1. Desarrollo de ejercicios en Lenguaje Haskell
2. Desarrollo de ejercicios en Lenguaje Java
3. Desarrollo de ejercicios de demostración de algoritmos, sobre implementaciones concretas de problemas tipo, usando la herramienta Key-Hoare Tool.

Prácticos de máquina:

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

VIII - Regimen de Aprobación

CONDICIONES PARA DICTADO REGULAR

- 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.
Se tomarán dos recuperaciones para cada parcial, acorde a la Ord.32/14 CS.

- Para promocionar la asignatura:
el alumno debe cumplir con las condiciones de regularización y aprobar los exámenes parciales o sus correspondientes recuperaciones, presentar y aprobar en tiempo y forma los prácticos de máquina solicitados por la cátedra y la evaluación integradora con una nota superior o igual a 7.
Se debe asistir al 80% de las clases teóricas y de las clases prácticas.
Se tomarán dos recuperaciones para cada parcial, acorde a la Ord.32/14 CS.

CONDICIONES EXCEPCIONALES - Programa en Fase No Presencial
Aula Virtual en: https://www.evirtual.unsl.edu.ar/moodle/
Método de acceso estudiantil: matriculación con contraseña

Se trabajará de la siguiente manera:
1) Modalidad Asincrónica
En el aula virtual se publicará material obligatorio y opcional para poder desarrollar las habilidades básicas necesarias para poder regularizar o promocionar la materia. En particular: guías de estudio, bibliografía obligatoria y opcional, videos de clases teórico/prácticas, prácticos y laboratorios de entrega obligatoria.

2) Modalidad Asincrónica con límite temporal
Realizar Evaluaciones diagnósticas on-line en los días estipulados por la Cátedra.
Presentar los laboratorios solicitados los días estipulados por la Cátedra enviándolos por mail a la cuenta: progUNSLCMN@gmail.com

3) Modalidad Sincrónica
Los alumnos deberán
• conectarse vía Internet en los horarios de asistencia ALTAMENTE RECOMENDADA a prácticos. Para tal fín se usará la aplicación gratuita ZOOM para realizar videoconferencias con el equipo de Cátedra a los fines de estimular el trabajo colaborativo, promover discusiones de pares y plantear diferentes posibilidades de desarrollo, en forma supervisada y coordinada por los docentes. Los alumnos serán consultados sobre sus posibilidades de conectividad, a los fines de proveer soluciones alternativas para aquellos alumnos que no puedan cumplir esta modalidad. Se hará registro de asistencias a dichas clases y de la respuesta por parte de los alumnos a las soluciones alternativas consensuadas con quienes no puedan cumplir esta modalidad.
• conectarse vía Internet en los horarios de asistencia RECOMENDADA a clases teórico/prácticas interactivas, usando la aplicación mencionada en el punto anterior. Las clases teóricas serán grabadas y publicadas posteriormente en el aula virtual, a los fines de poder ser accedidas de manera asincrónica.
• conectarse vía Internet para el desarrollo de las evaluaciones parciales y sus respectivas recuperaciones solicitadas por la cátedra. Los alumnos serán consultados sobre sus posibilidades de conectividad, a los fines de consensuar soluciones alternativas con aquellos alumnos que no puedan cumplir esta modalidad

IX - Bibliografía Básica

[1] Guía de estudio del Lenguaje Haskell. Material provisto por la Cátedra.
[2] Learn You a Haskell for Great Good!.A Beginner's Guide by Miran Lipovaca. No Starch press. April 2011, 400 pp. ISBN-13: 978-1-59327283-8, ISBN-10: 978-1-59327283-8. Adaptación al español gratuita on-line en http://aprendehaskell.es/).
[3] Real World Haskell, Bryan O'Sullivan, Don Stewart, and John Goerzen. Paperback: 700 pages, O'Reilly, November 2008, English, ISBN-10: 0596514980, ISBN-13: 978-0596514983. Disponible on line gratuito en http://book.realworldhaskell.org/
[4] Thinking in Java (4th Edition),Bruce Eckel, Prentice Hall; 4 edition (February 20, 2006), ISBN-10: 0131872486, ISBN-13: 978-0131872486.
[5] Aprender a programar Java desde cero. Autor: Mario Rodríguez Rancel. 200 páginas; Editorial: aprenderaprogramar.com, 2012. ISBN: 978-84-939427-0-0.

[6] A Discipline of Programming, Edsger W. Dijkstra, Prentice Hall, Inc. (October 28, 1976), ISBN-10: 013215871X, ISBN-13: 978-0132158718

X - Bibliografia Complementaria

[1] Haskell: The Craft of Functional Programming (2nd Edition), Simon Thompson, Addison-Wesley, ISBN 0201882957, 2011.
[2] Introduction to programming in Java. Hume, J.N. Patterson; Stephenson, Christine; Holt Software Assoc Inc. 2000. ISBN: 0921598394.
[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
[4] Logic in Computer Science modelling and reasoning about systems, Michael Huth and Mark Ryan; 427 pages (2nd edition). Cambridge University Press (link) in paperback only: ISBN 0 521 54310X

XI - Resumen de Objetivos

1. Estudio y aplicación del paradigma de la prog. orientada a objetos.
Lenguaje Java
2. Estudio y aplicación del paradigma de la programación funcional.
Lenguaje Haskell
3. Estudio y aplicación de técnicas de verificación de software,
en el estilo de demostración semántico funcional de Dijkstra, basado en la lógica axiomática de Hoare.

XII - Resumen del Programa

1. El paradigma de la Programación Orientada a Objetos:
Objetos. Mensajes. Clases. Herencia. Sobrecarga y Polimorfismo. Excepciones

2. El paradigma de la Programación Funcional:
Definición y manipulación de funciones simples y de orden superior. Definición y manipulación de tipos de datos polimorficos y recursivos. Recursión estructural. Recursion de cola.

3. Verificación de Programas:
Precondición más débil (weakest precondition WP). Determinismo y disyuntividad. Iteraciones. Aserciones, Invariantes.

XIII - Imprevistos

XIV - Otros