

Facultad de Ingeniería - Universidad Nacional de Cuyo			
P1- PROGRAMA DE ASIGNATURA			
Asignatura:	Paradigmas de Programación		
Profesor Titular:	Pablo Javier Vidal		
Carrera:	Licenciatura en Ciencias de la computación		
Año: 2018	Semestre: 4to	Horas Semestre: 112	Horas Semana: 7

OBJETIVOS

- ♦ Reconocer los fundamentos, características y diferencias de los distintos paradigmas de programación que son utilizados por los lenguajes de programación actuales.
- ♦ Identificar el paradigma que resulta más adecuado de aplicar para resolver diferentes tipos de problemas.
- ♦ Reconocer el modelo formal o semiformal subyacente de cada paradigma y la forma en que el mismo es incorporado en un lenguaje de programación correcto
- ♦ Adquirir las nociones básicas de los modelos de programación orientada a objetos, lógico y funcional.
- ♦ Aplicar los diferentes paradigmas en la resolución de problemas.

CONTENIDOS

UNIDAD 1: PARADIGMAS DE PROGRAMACIÓN

1.1. Concepto de paradigma de programación. Necesidad de la existencia de diferentes paradigmas de programación. Diferencia entre lenguaje y paradigma de programación.

1.2. Concepto de tipo: representación de los tipos en los diferentes lenguajes de programación. Importancia del concepto de tipo en relación a la implementación de sistemas complejos y cambiantes. Comparación de los diferentes esquemas de chequeo de tipos. Ubicación de los mecanismos de control de flujo en un programa.

1.3. Declaratividad: importancia de la separación del control de flujo de la lógica del dominio a modelar. Abstracción y modularización: definición y mecanismos de implementación. Concepto de ligaduras. Tipos de Ligaduras.

1.4. Orden superior: concepto e implicancias en el desarrollo de programas. Utilización de las variantes del polimorfismo en los diferentes paradigmas. Comparación entre los diferentes paradigmas de programación.

UNIDAD 2: PARADIGMA ORIENTADO A OBJETOS

2.1. Conceptos del modelo orientado a objetos. Concepto de Objeto. Concepto de mensaje, Centro Universitario (M5502KFA), Ciudad, Mendoza. Casilla de Correos 405. República Argentina.
Tel. +54-261-4494002. Fax. +54-261-4380120. Sitio web: <http://fing.uncu.edu.ar>

estado y comportamiento. Encapsulamiento. Visión de programa entendido como un conjunto de objetos que envían mensajes. Ambientes de objetos: diferencia con la programación tradicional.

2.2. Objetos. Constructores y destructores. Métodos y mensajes. Los métodos como mecanismo de resolución de mensajes.

2.3. Clases. Concepto de Clase como modelo/molde de objetos. Delegación y responsabilidad. Concepto de referencia. Interfaz e implementación: encapsulamiento del estado interno, ocultamiento de datos. Tipos de mensaje.

2.4. Áreas de aplicación. Lenguajes O.O. Relaciones entre objetos. Asociación, agregación y composición. Herencia. Herencia simple y múltiple. Realización, instanciación/dependencia. Variables y métodos de clase. Igualdad e identidad. Relaciones entre clases: asociación, composición; relación con delegación. Polimorfismo.

2.5. Implementación de programas O.O. en Java. Características generales, sintaxis y semántica del lenguaje. Estructura de programas. Clases. Estructura. Instanciación y destrucción de objetos. Variables y métodos, de instancia y de clase. Sobrecarga de métodos. Calificadores de miembros: de acceso y almacenamiento. Packages. Estructura. Exportación e importación. Herencia. Autoreferencia y referencia a superclases. Subclases. Redefinición de métodos. Clases abstractas. Clases parametrizadas. Clases anidadas. Interfaces. Formas de herencia. Persistencia de objetos. Contenedores de objetos. Eventos. Excepciones. Jerarquía de herencia. Lanzamiento, captura y tratamiento. Entidades y ligaduras.

2.6. Concurrencia. Concepto. Procesos e hilos de ejecución. Monitores.

UNIDAD 3: PARADIGMA LÓGICO

3.1. Conceptos de programación lógica. Fundamentos del modelo. Áreas de aplicación.

3.2. Propositiones lógicas. Lógica simbólica. Cálculo de predicados. Lógica de predicados de primer orden. Formas proposicionales. Cláusulas de Horn. Lógica con restricciones. Principales modos de inferencia. Principio de resolución. Métodos razonamiento de la lógica de predicados.

3.3. Implementación de programas en Prolog. Entorno de trabajo. Características del lenguaje. Sintaxis y semántica del lenguaje. Estructura general de programas. Realización de consultas.

3.4. Variables. Objetos elementales. Operadores. Metas. Hechos y reglas. Variables anónimas. Instanciación y unificación. Tratamiento de listas. Backtracking. Búsqueda de objetivos. Operador de corte. Functores.

UNIDAD 4: PARADIGMA FUNCIONAL

4.1. Conceptos de programación funcional. Fundamentos del modelo. Áreas de aplicación.

4.2. Concepto matemático de función. Definición por comprensión y por extensión. Definiciones y cálculo funcional. Notación y cálculo lambda. Expresiones funcionales y evaluación tardía.

4.3. Implementación de programas en Haskell. Características del lenguaje. Entorno de trabajo, definición de programas. Sintaxis y semántica del lenguaje. Estructura general de

programas. Secciones y guiones. Uso del intérprete. Entidades y ligaduras.

4.4. Evaluación de expresiones. Tipos de datos. Variables. Operadores infijos y prefijos. Expresiones. Notación bidimensional. Módulos. Notación de listas. Funciones. Tipos de funciones. Especificaciones funcionales: lambda, con guardas, con patrones. Parámetros: evaluación diferida. Retorno de valores. Reglas de precedencia.

4.5. Funciones primitivas. Clasificación. Recursividad. Tuplas y listas. Operadores. Tipos de listas. Enunciados type y data.

4.6. Comparación Paradigma Imperativo.

METODOLOGÍA DE ENSEÑANZA

Por tratarse de una asignatura con importante respaldo conceptual, y para facilitar la apropiación de los conocimientos, se aplicará la ejemplificación y demostración de manera metódica. Los contenidos del programa se presentan y analizan en las clases teóricas y se consolidan con los trabajos prácticos. Existe una estrecha relación entre la teoría y práctica, a su vez la práctica fundamenta sus procedimientos en la teoría.

Los ejercicios están orientados al análisis o entendimiento de las soluciones propuestas y a la concepción o diseño para responder a nuevos requerimientos y problemas. Para ello el material de práctica es seleccionado de manera de cubrir un amplio espectro temático y muestra diferentes niveles de dificultad. Algunos ejercicios son resueltos completamente en clase, y otros son encaminados para ser completados por los alumnos fuera de clase.

Estrategias metodológicas y de enseñanza

Las siete horas semanales que tiene asignada la materia se distribuyen de la siguiente manera:

- Clases teóricas de tres horas para presentación y exposición de los temas, con el uso de pizarrón con apoyo de proyección de imágenes y computadora. Dichas clases incluyen gran contenido de práctica y serán participativas y de debate. Se desarrollarán de manera expositiva los conceptos fundamentales, con presentación de casos de estudio prácticos de aplicación inmediata del tema presentado. Se promoverá la investigación y la búsqueda de información por parte del alumno. Se exigirá un trabajo continuo al alumno tanto dentro del aula como fuera de ella.
- Clases prácticas de cuatro horas de ejercitación para resolución de problemas y consultas trabajando en el aula. Estas clases incluirán presentación del tema, casos de estudio abiertos a debate y de construcción colectiva, y consulta individual de ejercicios. Se incluyen actividades de laboratorio para aplicar diferentes lenguajes de programación (Java, Prolog, Haskell) como herramienta instrumental de resolución de problemas de acuerdo al paradigma que se esté estudiando.

Actividad	Carga horaria por semestre
Teoría y resolución de ejercicios simples	48
Formación práctica	
Formación Experimental – Laboratorio	64
Formación Experimental - Trabajo de campo	0
Resolución de problemas de ingeniería	0
Proyecto y diseño	0
Total	112

BIBLIOGRAFÍA

Bibliografía básica

Autor	Título	Editorial	Año	Ejemplares en biblioteca
T.W. Pratt, M.V. Zelkowitz	Programming Languages - Design and Implementation, 4 ^o edición	Prentice Hall	2000	
Deitel&Deitel	Como Programar en Java, 7ma edición	Prentice Hall	2008	
R. Bird	Introducción a la Programación Funcional con Haskell	Prentice Hall	2000	
W.D. Burnham, A.R. Hall	Prolog, Programación y Aplicaciones	Limusa	1990	
Allen B. Tucker, Rober E. Noonan	Programming Languages_ Principles and Paradigms	McGraw-Hill	2007	

Bibliografía complementaria

Autor	Título	Editorial	Año	Ejemplares en biblioteca
John S. dean, Raymond H. Dean	Introducción a la Programación con Java	McGraw-Hill	2009	
H. Schildt	Java 7 - Oracle Press	ANAYA	2012	

EVALUACIONES (S/ Ord. 108-10_CS)

Se realizará una evaluación integral y continua durante el cursado. **Se registrá por la Ord. 108/10 C.S. UNCUYO**, explicitando claramente en la planificación didáctica del Espacio Curricular el régimen de acreditación y promoción, según Artículos. 7-8 y 9 de la mencionada norma. Se tendrá en cuenta en la evaluación las características de validez, pertinencia y consistencia con la propuesta de enseñanza y aprendizaje realizada.

Evaluaciones durante el cursado

Para la evaluación continua en el aula se considerarán la resolución de problemas, trabajos prácticos y evaluaciones parciales, con observación del desempeño individual y grupal y retroalimentación constante. La evaluación incluirá propuestas de autoevaluación que posibiliten la reflexión del alumno como protagonista de su proceso de aprendizaje.

Condición de regularidad tras el cursado

Para adquirir la condición de alumno regular se deberá cumplir con los siguientes requisitos:

- Aprobar los tres parciales, cada uno de ellos consta de puntaje máximo de 10 (DIEZ). Para la obtención de la regularidad es necesario aprobar las tres evaluaciones con un mínimo de 6 (SEIS) en cada una de ellos.
- Aquellos alumnos que desapruében cualquiera de las evaluaciones parciales o las tres, podrán rendir un examen recuperatorio global, con los mismos temas del examen parcial que desaprobó. El examen recuperatorio global se aprueba con 6 (SEIS) como mínimo sobre un total de 10 (DIEZ).
- *El Alumno Promocionado* será considerado aquel que obtenga la calificación en cada examen parcial igual o superior a 8 (OCHO). Un alumno que haya desaprobado algún examen parcial, no tiene opción a promoción.
- Las evaluaciones son teóricas prácticas, y cada examen quedará como respaldo documental de lo elaborado por el alumno, posibilitando cualquier revisión posterior que pueda ser necesaria por el mismo.

Evaluación final

Para la aprobación de la asignatura del estudiante en condición regular, se realizará una evaluación calificativa que apunta a identificar cuánto ha aprendido un alumno en este espacio curricular con el fin de tomar la decisión de acreditación de saberes. Constituye una evaluación escrita que refleje el trabajo del estudiante, comprensivo y fundamentado, y brinda al docente, una mirada global del proceso de aprendizaje.

Examen final para alumnos libres.



El mismo tiene dos fases, la primera de ellas es un examen escrito y se aprueba con un mínimo de 60 puntos sobre un total de 100. Una vez aprobado el examen escrito, el alumno libre deberá rendir un examen oral según los criterios de evaluación mencionados previamente.

Alumnos recursantes.

No hay régimen especial para alumnos recursantes.

FECHA, FIRMA Y ACLARACIÓN TITULAR DE CÁTEDRA