UNIVERSIDAD TECNOLÓGICA NACIONAL
Facultad Regional Resistencia
TECNICATURA SUP. EN PROGRAMACIÓN

French 414
Tel 03722-432928 TelFax: 03722-432683
(3500) Resistencia - Chaco

PLANIFICACIÓN DE CÁTEDRA

Ciclo Lectivo: 2019

Carrera: Tecnicatura Universitaria en Programación

Asignatura: PROGRAMACION II

Nivel: PRIMERO – SEGUNDO CUATRIMESTRE

Coordinador de la Carrera

Nombre y Apellido: Ing Germán Gaona e-mail: tsp@frre.utn.edu.ar

Docente:

Nombre y Apellido: FIGUEREDO, FERNANDO ARIEL
Categoría docente: Prof. Adjunto Interino
e-mail: figueredofernandoa@gmail.com

UNIVERSIDAD TECNOLÓGICA NACIONAL
Facultad Regional Resistencia
TECNICATURA SUP. EN PROGRAMACIÓN

French 414
Tel 0362-4432928 TelFax: 0362-4432683
(3500) Resistencia - Chaco

PLANIFICACIÓN DE CÁTEDRA

OBJETIVOS GENERALES DE LA ASIGNATURA

Lograr que los alumnos conozcan los principios básicos de la resolución de problemas.
Preparar los esquemas mentales para la resolución de problemas.
Aprender un lenguaje de programación de alto nivel.
Reconocer y emplear los conceptos básicos de la programación orientada a objetos.
Identificar cuestiones de ingeniería de software que aplican a grandes sistemas.
Mantener una cultura de trabajo en equipo.

PROGRAMA ANALÍTICO

UNIDAD 1 – INTRODUCCIÓN

Contenidos:

Complejidad Algorítmica

- Algoritmia
  - Fases en el desarrollo de un algoritmo.
  - Verificación y Análisis de Algoritmos.
- Eficiencia de Algoritmos
  - Tiempo de ejecución y uso de memoria.
  - Comportamiento asintótico
  - Funciones de complejidad en tiempo más usuales.

Subalgoritmos o Subprogramas

- Funciones
- Procedimientos
- Ámbito de variables Locales y Globales
- Paso de Parámetros
  - Paso por Valor.
  - Paso por Parámetro.

Recursividad

- Concepto y Tipos de Recursividad.
- Cuando usar y cuándo no usar Recursividad.

UNIDAD 2 – PROGRAMACIÓN ORIENTADA A OBJETOS

Contenidos:

Fundamentos de la Programación Orientación a Objetos.

- Clases y Objetos
  - Definición de clases. Instanciación de objetos. Diferencias.
  - Implementación de tipos de relaciones entre clases.
  - Definición del ciclo de vida de clases.
  - Definición del ciclo de vida de objetos.
  - Creación de variables de referencia y asignaciones.
- Constructores

French 414
Tel 0362-4432928 TelFax: 0362-4432683
(3500) Resistencia - Chaco

¿Qué es un constructor? Definición de constructor por defecto y constructores parametrizados.

Propiedades

Implementación de getters y setter para encapsular atributos de clases.

Modificadores de acceso

Definición de public, private y protected.

Pilares de la Orientación a Objetos

Herencia

Definición de herencia entre clases. Herencia Simple.

Polimorfismo

Implementación de polimorfismo de métodos. Ejemplos

Sobrecarga

Implementación de sobrecarga de métodos. Ejemplos.

Conceptos avanzados

Clases Abstractas e Interfaces

Implementación de clases abstractas.
Implementación de interfaces.
Usos comunes de ambas. Ejemplos.
Diferencias entre clases abstractas e interfaces.

Paquetes y Namespaces

Comprender la agrupación de clases por funcionalidades.
Agrupaciones lógicas y físicas de clases.
Importar funcionalidades de otras clases a través de paquetes o namespaces.

Aplicación de la Programación Orientada a Objetos

Resolver escenarios de la vida real a través del modelado de clases.
Entender el concepto de contextos.

Unidad 3 – TAD Y ESTRUCTURAS LINEALES

Contenidos:

Clasificación. Listas particulares y generalizadas. Organización y acceso.
Concepto de restricción. Noción de puntero.
Operaciones con Listas: recorrido, inserción, borrado.
Tipos Abstractos de Datos (TAD) y Diseño Orientado a Objetos.

Tipos de datos, estructuras de datos y tipos abstractos de datos.

Especificación de TADs.

El TAD conjunto.

Principios del diseño Orientado a Objetos.

Abstracción.
Encapsulamiento.
Modularidad.

Estructuras de Datos Dinámicas.

TAD Lista

Especificación formal del TAD Lista.

Implementación del TAD Lista con estructuras estáticas.
Implementación del TAD Lista mediante variables dinámicas.
Listas Doblemente Enlazadas.
Lista Circular mediante variables dinámicas.

TAD Pila

Especificación formal del TAD Pila.

Implementación del TAD Pila con arreglos.
Implementación del TAD Pila mediante variables dinámicas.

Aplicación de Pilas

Llamada a subprogramas o procedimientos.

Recursión.
Ordenamiento.

TAD Cola
Especificación formal del TAD Cola
Implementación del TAD Cola con arreglos.
Implementación del TAD Cola con Listas enlazadas y circulares.
Colas de Simulación.

Unidad 4 – TAD ARBOL

Implantación de Árboles N-arios.
Transformar árboles n-arios a binarios.
Árboles Binarios
Definiciones recursivas.
Características de los Árboles: profundidad, niveles, nodos.
Implementación de Árboles Binarios.
Recorrido de un árbol binario.
Reconstrucción de árboles binarios a partir de dos de sus recorridos: Anchura y Profundidad.
Tipos de recorridos: pre-orden, in-orden o post-orden.
Árbol binario de búsqueda.

Implementación de Árboles binarios con vectores.

Unidad 5 – TDA GRAFO Y TABLAS HASH

Conceptos y Definiciones relacionadas con Grafos
Representación de los Grafos.
Matriz de Adyacencia.
Lista de Adyacencia.
Recorridos de un Grafo.
Recorrido en Anchura o Amplitud.
Recorrido en Profundidad.
Algoritmos fundamentales con Grafos
Algoritmo de Dijkstra.
Algoritmo de Floyd.
Tablas Hash
Estructura interna de una Tabla Hash.
Estrategias de soluciones de Colisiones.

ESTRATEGIAS METODOLÓGICAS

Desarrollo de la Asignatura.

Las instancias para el desarrollo de la asignatura serán clases teóricas y prácticas. Inicialmente se tratará de familiarizar al alumno con los conceptos básicos que se deben dominar para poder acceder al entendimiento de las bases de la programación. Para ello es necesario que el alumno adquiera una primera aproximación a la resolución de problemas, especialmente a aquellos del tipo algorítmico.

Dinámica del Dictado de las Clases.

El dictado de la cátedra será de carácter teórico y práctico, estableciéndose en primer lugar los conceptos teóricos seguido de una aplicación inmediata por medio de estudio dirigido y trabajos prácticos sobre las PCs. Se estimulará el trabajo grupal, enfatizando el desarrollo de los ejercicios prácticos. El Profesor procederá a describir técnicas, características y pondrá ejemplos. Asimismo habrá una parte de la clase dedicada a la aplicación de los conceptos vistos durante la parte teórica. Se desarrollarán problemas con creciente nivel de dificultad.

Ejercicios Prácticos.

Las actividades se plantean con presentación y resolución de problemas de baja complejidad con el objetivo de reafirmar conceptos vistos en teoría. En los laboratorios se trabajará sobre una guía de trabajos prácticos que abarcan los temas vistos en teoría. Los alumnos podrán resolver los ejercicios de la guía y revisar sus soluciones en clases con el Profesor, de esta manera se realizará una evaluación continua del progreso de los alumnos, estimulando además la autoevaluación por parte del estudiante, como medio idóneo para aumentar su rendimiento.

CRONOGRAMA:

CLASE Nº / FECHA	CONTENIDO A SER DESARROLLADO
Semana 1 /11-03-2019	UNIDAD 1
Semana 2 /18-03-2019	UNIDAD 1
Semana 3 /25-03-2019	UNIDAD 1
Semana 4 /01-04-2019	MESAS DE EXAMEN
Semana 5 /08-04-2019	UNIDAD 2
Semana 6 /15-04-2019	UNIDAD 2
Semana 7 /22-04-2019	UNIDAD 3
Semana 8 /29-04-2019	UNIDAD 3
Semana 9 /06-05-2019	PRIMER PARCIAL
Semana 10 /13-05-2019	UNIDAD 3
Semana 11 /20-05-2019	UNIDAD 4
Semana 12 /27-05-2019	UNIDAD 4
Semana 13 /03-06-2019	UNIDAD 5
Semana 14 /10-06-2019	UNIDAD 5
Semana 15 /17-06-2019	SEGUNDO PARCIAL
Semana 16 /24-06-2019	PRIMER RECUPERATORIO
Semana 17 /01-07-2019	SEGUNDO RECUPERATORIO

ORGANIZACIÓN DE ESPACIOS DENTRO Y FUERA DEL ÁMBITO UNIVERSITARIO

Para las clases se requerirá un laboratorio de computación que permita la utilización de proyector y que permita la conformación de grupos de trabajo.

RECURSOS NECESARIOS:

Laboratorio de Computación con proyector y conexión a Internet.
Software de desarrollo instalado en la PCs

- JAVA Development KIT.
- JAVA SDK (JVM)
- IntelliJ
- NetBeans (opcional)
- Eclipse (opcional)

EVALUACIÓN: INSTRUMENTOS Y MODALIDAD- RÉGIMEN DE APROBACIÓN

La aprobación del cursado de la asignatura se basa en:

APROBACIÓN DE LA CURSADA (REGULARIDAD)

Las condiciones para regularizar la materia son:

- 75% de asistencia a las clases teóricas y prácticas.
- El alumno que no alcance los objetivos de aprobación directa, estará habilitado a rendir evaluación final.

APROBACIÓN DIRECTA

Las condiciones para aprobar directamente la materia son:

- 75% de asistencia a las clases teóricas y prácticas
- Aprobar las instancias de evaluación teóricas, prácticas y/o ambas instancias con nota mínima de seis (6) puntos.
- En el caso en que no haya aprobado alguna de las instancias de evaluación parcial, tendrá al menos una instancia de recuperación.
- La nota promedio de las instancias de evaluación así obtenida será la calificación definitiva de aprobación de la asignatura.

CORRELATIVIDADES

Para Cursar

Cursada

Programación I
Laboratorio de Computación I

Aprobada

Ninguna

Para Rendir

Aprobada

Programación I
Laboratorio de Computación I

.

BIBLIOGRAFÍA

GOODRICH y TAMASSIA. Estructura de Datos y Algoritmos en Java. CECSA. 2002.
JOYANES AGUILAR, LUIS. Programación Orientada a Objetos. McGraw-Hill. 2002.
JOYANES AGUILAR, LUIS, Algoritmo y Estructura de Datos. McGraw-Hill. 1998.

BIBLIOGRAFÍA DE CONSULTA

JOYANES AGUILAR, LUIS, Fundamentos de Programación: Algoritmos y Estructura de Datos. Mcgraw-Hill. 1996.
ALLEN WEISS, MARK. Estructura de Datos y Algoritmos. Addison-Wesley Iberoamericana. 1995.
AHO, ALFRED V; HOPCROFT, JOHN E; ULLMAN, JEFFRY D. Estructura de Datos y Algoritmos. Addison-Wesley Iberoamericana. 1990.
CAIRO, OSVALDO. Estructura de Datos. McGraw-Hill. 2002.
HEILEMAN, GREGORY. Estructura de Datos, Algoritmos y Programación orientada a objetos. McGraw-Hill. 1998.
RUSSEL L. Introduction to Computing and Algorithms. Addison-Wesley. 1998.
AHO, ALFRED. Estructura de datos y algoritmos. Addison-Wesley. 1988. OBJECT-ORIENTED ANALYSIS AND DESIGN WITH APLICATIONS. BLOOCH, GRADY. SECOND EDITION, THE BENJAMIN/CUMMINGS PUBLISHING COMPANY, INC. 1994. OBJECT-ORIENTED MODELING AND DESIGN. JAMES RUMBAUGH ET AL. PRENTICE-HALL, INC.
1991.
ARNOLD, GOSLING, HOLMES. Holmes. El lenguaje de programación Java. Addison-Wesley. 3ra. Edición 2002.
ECKEL, BRUCE. Pensando en Java. Prentice-Hall. 2006.