

Facultad de Ingeniería - Universidad Nacional de Cuyo			
P1- PROGRAMA DE ASIGNATURA			
Asignatura:	Informática		
Profesor Titular:	Mg. Lic. Javier J. Rosenstein		
Carrera:	Ingeniería Mecatrónica		
Año: 2022	Semestre: 6	Horas Semestre: 60	Horas Semana: 4

OBJETIVOS

Establecer con criterio los ámbitos de aplicación de sistemas de cómputo basados en PC, microcontroladores y arquitecturas especiales, comprendiendo tanto el soporte físico como lógico. Familiarizarse con los sistemas operativos para PC, lenguajes de programación de mayor uso y entornos de desarrollo y depuración de software. Introducir a los lenguajes procedurales de alto nivel para resolución de problemas de cómputo y automatismos, mediante los enfoques Diagrama de Flujo, Diagrama de Estados y Estructurado. Utilizar con solvencia los tipos de datos simples y estructurados.

CONTENIDOS

UNIDAD 1: Introducción

1.A. Arquitectura de computadoras

Introducción a la arquitectura de computadoras. Organización de sistemas de cómputo en capas. Jerarquía, lenguajes y niveles de abstracción. Sistemas de numeración. Introducción al manejo de punto flotante para cálculos numéricos.

1.B. Sistemas Operativos

Introducción a los Sistemas Operativos: conceptos y roles. Componentes: procesos, manejador de memoria, entrada/salida.

1.C. Introducción a los Lenguajes de programación

Lenguaje de programación: conceptos, niveles de abstracción. Paradigmas de programación. Lenguajes de propósito general y de propósito particular. Interpretación vs. Compilación. Lenguajes fuerte y débilmente tipados. El proceso de la compilación, enlazado y ejecución: programa fuente, programa objeto, programa ejecutable. Entornos Integrados de Desarrollo (IDE): entornos modernos más utilizados, uso básico, resaltado de sintaxis, navegación de código, depuración (traza, inspección de variables).

UNIDAD 2: Introducción a la programación

2.A Lenguajes compilados fuertemente tipados: Introducción al lenguaje C

Características principales. Estructura general de un programa en C. Componentes del lenguaje: sintaxis básica, identificadores (variables y funciones), constantes, operadores, declaración de variables. Preprocesador: concepto, directiva `#include`, directiva `#define`,

otras directivas.

2.B. Tipos de datos, Operadores, Expresiones Y Sentencias

Tipos de datos primitivos. Operadores unarios y binarios. Operadores aritméticos. Operadores lógicos. Operadores a nivel de bits. Expresiones aritméticas, lógicas, generales. Reglas de precedencia y asociatividad. Conversiones de tipo implícitas y explícitas (casting).

2.C. Control básico del flujo de ejecución

Sentencias simples, bloques de sentencias. Estructuras condicionales: sentencia if/else/else if. Selector de caso. Estructuras de iteración: sentencia while, sentencia for, sentencia do/while. Estructuras de control anidadas. Uso práctico del depurador.

2.D. Subrutinas

Conceptos generales: cohesión, acoplamiento, encapsulamiento y reutilización de código. Utilidad de las funciones/procedimientos. Definición de la cabecera (signature) de una función/procedimiento: nombre, argumentos y valor de retorno. Llamada a una función. Variables: alcance, vida y visibilidad de las variables. Recursión.

Unidad 3: Estructuras de datos e interacción con el sistema operativo

3.A. Estructuras de datos

Arreglos: vectores, cadenas de caracteres, arreglos multidimensionales. Tipos definidos por el usuario. Tipos de datos estructurados.

3.B. Apuntadores

Conceptos de apuntador y direccionamiento. Operadores de dirección (&) e indirección (*). Pasaje de argumentos a funciones por valor y por referencia. Declaración de apuntadores. Apuntadores múltiples, violaciones de segmento, fugas de memoria. Relación entre arreglos y apuntadores. Manejo de memoria dinámica. Estructuras de datos dinámicas: listas enlazadas, pilas, colas, grafos y árboles.

3.C. Interacción con el sistema operativo

Llamadas al sistema. Hilos (threads). Entrada/Salida: tipos de archivos, operaciones básicas, tuberías (pipes).

UNIDAD 4: Resolución de problemas algorítmicos

4.A. Introducción a la resolución de problemas algorítmicos

Problema. Contexto. Datos asociados. Comprensión de problemas y metodología general de resolución. Definición de algoritmo y características. Definición de Programa. Pseudocódigo. Diseño de estructuras de datos.

4.B. Lenguajes interpretados dinámicos: Introducción al lenguaje Python

Características principales. Estructura general de un programa en Python. Componentes del lenguaje: sintaxis básica, identificadores (variables y funciones), tipos de datos básicos, constantes, operadores, declaración de variables. Operadores y expresiones. Control de flujo de ejecución: sentencias condicionales y bucles. Subrutinas. Estructuras de datos. Modelo de gestión de memoria.

4.C. Aplicaciones

Búsqueda y ordenamiento en arreglos. Aplicaciones de estructuras de datos dinámicas (listas, árboles y grafos). Aplicaciones embebidas: uso de entradas/salidas analógicas/digitales, acceso a los recursos de las plataformas embebidas.

METODOLOGÍA DE ENSEÑANZA

Se empleará una metodología teórico-práctica. Se estima utilizar aproximadamente el 35% del tiempo para desarrollar los conceptos teóricos, y el 65% restante para desarrollar actividades prácticas, incluyendo mini-casos de estudio, solución de problemas mecatrónicos (simulados) y un proyecto final integrador. Se espera además que los alumnos dediquen tiempo adicional a realizar ejercicios prácticos en tiempos fuera del aula.

Actividad	Carga horaria por semestre
Teoría y resolución de ejercicios simples	20
Formación práctica	
Formación Experimental – Laboratorio	10
Formación Experimental - Trabajo de campo	0
Resolución de problemas de ingeniería	20
Proyecto y diseño	10
Total	60

BIBLIOGRAFÍA

Bibliografía básica

Autor	Título	Editorial	Año	Ejemplares en biblioteca
A. Tanenbaum	Sistemas Operativos Modernos, 3° Ed.	Pearson	2009	0
A. Tanenbaum	Organización de Computadoras: Un enfoque estructurado, 4° Ed.	Prentice-Hall	2000	0
A. Tanenbaum	Structured Computer Organization, 5th ed.	Prentice-Hall	2005	0
Python Software Foundation	The Python Tutorial. https://docs.python.org/3/tutorial/		2018	0
L. Joyanes Aguilar	Fundamentos de Programación, 4ta ed.	McGraw Hill	2008	0
L. Joyanes Aguilar, I. Zahonero Martínez	Programación en C : metodología, algoritmos y estructura de datos	McGraw-Hill	2005	1
O. Cairó Battistutti,	Metodología de la programación : algoritmos, diagramas de flujo y programas	Alfagrama	2003	1
B. Kernighan, D. Ritchie	El lenguaje de programación C	Prentice-Hall	1991	1
B. S. Gottfried	Programación en C	McGraw-Hill	2005	1

Bibliografía complementaria

Autor	Título	Editorial	Año	Ejemplares en biblioteca
D. Robbins	POSIX threads explained	IBM Developer Works Technical Library	2000	0
D. Robbins	Common threads: POSIX threads explained (Part 2)	IBM Developer Works Technical Library	2000	0
D. Robbins	Common threads: POSIX threads explained (Part 3)	IBM Developer Works Technical Library	2000	0
W. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery	Numerical Recipes in C	Cambridge University Press	1997	0
W. Bolton	Ingeniería de Control, 2° ed.	Alfaomega	2001	1

EVALUACIONES (S/ Ord. 108-10_CS)

- Para aprobar o regularizar la materia, se requiere la asistencia al 80% de las clases como mínimo
- Los exámenes incluyen 2 parciales escritos, y un coloquio final integrador
- Para regularizar la materia, además de la asistencia, se deberán entregar las guías de trabajos prácticos y de laboratorio correctamente resueltas y aprobar todos los exámenes (o sus correspondientes recuperatorios) con una nota de 6 puntos o más, de acuerdo con lo especificado en la Ordenanza 108/2010-CS.
- Para promocionar la materia, además de cumplir con los requisitos mínimos exigidos para regularizarla, la nota final debe ser de 7 (siete) puntos o más, de acuerdo con la siguiente fórmula:

$$\text{Nota final} = \text{Parcial 1} * 0.30 + \text{Parcial 2} * 0.30 + \text{Trabajo Integrador y coloquio} * 0.40$$

Régimen especial para alumnos libres

- Desarrollar y presentar con 2 semanas de anticipación a la mesa de examen todos los Trabajos Prácticos que se desarrollan a lo largo de la cátedra.
- Una vez entregados los prácticos, desde la cátedra se le dará un trabajo especial integrador, el cual será diferente en cada mesa de examen.
- El trabajo especial, específico para la mesa de examen, debe ser entregado 1 semana antes de la mesa de examen para su evaluación previa
- En la mesa de examen se tomará un examen escrito, abarcando todos los temas del programa. Si se aprueba este examen escrito, se tomará un coloquio oral sobre el trabajo especial de la mesa, sobre todos los trabajos entregados, y sobre la teoría. La nota final será determinada en base a los exámenes oral y escrito mencionados



Programa de examen

- Examen parcial N° 1
- Examen parcial N° 2
- Exámenes recuperatorios.
- Entrega trabajo final integrador.
- Coloquio final integrador (oral)

FECHA, FIRMA Y ACLARACIÓN TITULAR DE CÁTEDRA