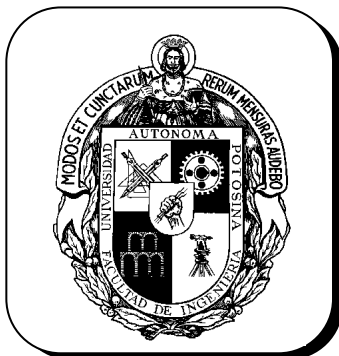


FACULTAD DE INGENIERÍA

CENTRO DE INVESTIGACIÓN Y ESTUDIOS DE POSGRADO



Nombre de la materia: TALLER DE PROGRAMACIÓN ESTRUCTURADA

Clave Facultad:

Clave U.A.S.L.P.:

No. de créditos: 4

Horas/Clase/Semana: 2

Horas totales/Semestre: 32

Horas/Práctica (y/o Laboratorio):

Prácticas complementarias:

Trabajo extra clase Horas/Semana: 2

Carrera/Tipo de materia: Posgrado en Ingeniería Mecánica
Básica obligatoria para Maestría

No. de créditos aprobados:

Fecha última de Revisión Curricular: Marzo 2020

JUSTIFICACIÓN DEL CURSO

Esta materia está dirigida a desarrollar el razonamiento lógico para la resolución de problemas, proporcionar una base firme para el conocimiento de lenguajes de programación acordes a la programación estructurada e introducir el uso del Lenguaje C en la solución de problemas relacionados con la ingeniería.

Adicionalmente se introducirá al alumno en la programación orientada a objetos, aplicando metodologías y técnicas que permitan un mejor manejo de los datos y una mejor organización del código.

OBJETIVO DEL CURSO

Desarrollar el razonamiento lógico necesario y aprender la sintaxis básica del Lenguaje de Programación C. Con estas habilidades el alumno podrá estructurar y desarrollar soluciones a problemas básicos de ingeniería

con el apoyo de la computadora, utilizando el paradigma de programación estructurada. Además, se busca el conocer y aplicar los conceptos principales de la programación orientada a objetos, y la creación de bibliotecas de clases.

CONTENIDO TEMÁTICO

1. INTRODUCCIÓN A LA PROGRAMACIÓN 1 hr

Objetivo: Que el alumno conozca los conceptos básicos relacionados con la computadora, plataforma utilizada para la realización de programas.

- 1.1. Categorías de computadoras
- 1.2. Partes de una computadora
- 1.3. Tipos de software
- 1.4. Introducción a la lógica booleana
- 1.5. Diagramas de flujo
- 1.6. Algoritmos

2. INTRODUCCIÓN AL LENGUAJE C 1 hr

Objetivo: Que el alumno conozca la estructura básica de un programa en C.

- 2.1. Historia de C
- 2.2. Estructura de un programa en C
- 2.3. Proceso de compilación y ligado

3. CONCEPTOS BÁSICOS 2 hrs

Objetivo: Que el alumno conozca los elementos básicos del lenguaje de programación C.

- 3.1. Tipos de datos
- 3.2. Variables
- 3.3. Constantes
- 3.4. Operadores y expresiones
- 3.5. Cadenas de caracteres
- 3.6. Entrada/salida por consola

4. ESTRUCTURAS DE CONTROL DE FLUJO 3 hrs

Objetivo: Que el alumno sea capaz de aplicar las estructuras de control de flujo en la solución de problemas de ingeniería y problemas en general.

- 4.1. Estructuras de selección
- 4.2. Estructuras de repetición
- 4.3. Sentencias break y continue

5. ARREGLOS	3 hrs	9. ESTRUCTURAS	3 hrs
Objetivo: Que el alumno sea capaz de almacenar datos en arreglos, con el fin de que sean utilizados en la solución de problemas.		Objetivo: Que el alumno conozca el uso de estructuras para el almacenamiento de datos relacionados entre sí.	
5.1. Introducción a los arreglos		9.1. Estructuras	
5.2. Arreglos unidimensionales		9.2. Estructuras como parámetros	
5.3. Arreglos bidimensionales		9.3. Arreglos de estructuras	
6. FUNCIONES	4 hrs	9.4. Apuntadores a estructuras	
Objetivo: Que el alumno sea capaz de estructurar programas utilizando funciones, con el fin de facilitar su mantenimiento e incrementar la reutilización de código.		10. RECURSIÓN	2 hrs
6.1. Formato		Objetivo: Que el alumno conozca el concepto de recursión, así como sus ventajas y desventajas.	
6.2. Definición y llamada		10.1. Concepto	
6.3. Paso de argumentos por valor		10.2. Ejemplos clásicos	
6.4. Ámbito de variables		11. DESARROLLO DE CÓDIGO	4 hrs
6.5. Funciones que regresan valores		Objetivo: Que el alumno conozca metodologías de desarrollo sistemática de código.	
6.6. Arreglos como parámetros		11.1. Desarrollo del diagrama de flujo	
7. LIBRERIAS	2 hrs	11.2. Implementación modular, “top-down”, “bottom-up”	
Objetivo: Que el alumno conozca las librerías básicas y como aprovecharlas en el desarrollo de un código.		11.3. Herramientas de prueba y búsqueda de errores (“debugging”)	
7.1. Uso de librerías de funciones en un código		11.4. Prueba y verificación modular	
7.2. Generación de librerías propias para uso compartido en proyectos		11.5. Documentación	
7.3. Librerías comunes y de licencia pública		11.6. Desarrollo de proyecto.	
8. MEMORIA DINÁMICA	3 hrs	12. INTRODUCCIÓN A OTROS AMBIENTES	4 hrs
Objetivo: Que el alumno sea capaz de utilizar apuntadores y memoria dinámica para el manejo eficiente de la memoria.		Objetivo: Que el alumno tenga una visión general de las opciones.	
8.1. Concepto de apuntador		12.1. Introducción a Programación Orientada a Objetos y el lenguaje C++	
8.2. Aritmética de apuntadores		12.2. MatLab: programación de manipulación con vectores y matrices de forma eficiente	
8.3. Manejo de memoria dinámica		12.3. Introducción a Programación Visual y desarrollo de interfaz gráfica (GUI)	
8.4. Paso de argumentos por referencia			
8.5. Arreglos y apuntadores			

METODOLOGÍA

El curso se enfoca a que el alumno desarrolle el razonamiento lógico mediante el análisis de problemas sencillos. Se hace énfasis en el proceso de resolución de problemas con programación, realizando inicialmente el análisis del problema, para posteriormente proceder a la elaboración de un diagrama de bloques o algoritmo, a partir del cual se implemente la solución en lenguaje C. A lo largo del curso se hace énfasis en la importancia del diseño para evitar errores y en el análisis del código con

el fin de detectar errores (prevención y corrección de errores).

La materia se imparte en un salón con pizarrón y cañón. En las sesiones se alternan actividades en las cuales se trabaja en el pizarrón y/o en papel, con actividades en las cuales se implementan en computadora algunos de los algoritmos desarrollados.

Se muestran ejemplos de aplicación y se realiza un proyecto con el fin de reforzar los conceptos.

EVALUACIÓN

De cada tema visto en clase se realizarán actividades y se dejarán trabajos de aplicación relacionados con la Ingeniería y otras disciplinas, de tal manera que para aprobar la materia sea requisito haber presentado por lo

menos el 80% de estos trabajos. Los trabajos y actividades constituirán el 20% de cada calificación parcial y el 80% lo constituirá el examen parcial.

BIBLIOGRAFÍA BÁSICA

Apuntes del curso

Deitel & Deitel. “Cómo programar en C/C++ y Java”, 6ª Edición. Ed. Pearson, 2008.

Joyanes Aguilar, Luis, Ignacio Zahonero Martínez, “Programación en C, C++, Java y UML”, McGraw-Hill Education, 2ª edición, 2014.

BIBLIOGRAFÍA COMPLEMENTARIA

Kernighan & Ritchie. “El lenguaje de programación C”, 2ª Edición, Prentice Hall, 1991.

<http://cplusplus.com/>