

1. CURSO

CS113. Programación II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS113. Programación II
2.2 Semestre	:	3 ^{er} Semestre.
2.3 Créditos	:	4
2.4 horas	:	3 HT; 3 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS112. Programación I. (2 ^{do} Sem) CS112. Programación I. (2 ^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este es el tercer curso en la secuencia de los cursos introductorios a la programación. En este curso se pretenden cubrir los conceptos relacionados con el concepto de Puntero en C y C++. El manejo de punteros nos permite realizar un acceso flexible a la memoria del computador consiguiendo elaborar programas eficientes. El dominio de este tema permitirá que los participantes puedan tener una buena base para elaborar soluciones computacionales a problemas de estructuras de datos avanzadas y algoritmos con manejo eficiente de memoria.

5. OBJETIVOS

- Introducir al alumno en los conceptos relacionados con Punteros en C y C++, permitiendo asimilar esta técnica de programación, la misma que es necesaria para desarrollar algoritmos y estructuras de datos eficientes.

6. RESULTADOS DEL ESTUDIANTE

- 1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (**Usar**)
- 3) Comunicarse efectivamente en diversos contextos profesionales. (**Usar**)
- 5) Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (**Usar**)

7. TEMAS

Unidad 1: Introducción a Punteros en C/C++ (5)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Declaración de punteros. • Trabajo con punteros: <ul style="list-style-type: none"> Referenciación. Desreferenciación. • Punteros tipados, aritmética de punteros, punteros void. • Punteros a punteros. • Punteros como argumentos de una función-llamada por referencia. 	<ul style="list-style-type: none"> • Introducir en el manejo de punteros, sus operadores y su interacción en la memoria.[Usar] • Demostrar mediante ejemplos los diferentes usos de los operadores con punteros.[Usar] • Demostrar mediante ejemplos el uso de aritmética de punteros. [Usar] • Demostrar mediante ejemplos, las diferentes llamadas a funciones y el uso de punteros. [Usar]
Lecturas : [Nak13], [stroustrup2013], [Reese2013], [Toppo2013]	

Unidad 2: Manejo de Punteros con arrays (5)	
Resultados esperados: 1,3	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Arrays como argumentos de una función. • Arrays de caracteres y punteros. • Punteros y Arrays de 2 dimensiones. • Punteros y arrays multidimensionales. 	<ul style="list-style-type: none"> • Demostrar el uso de punteros con diferentes tipos de Arrays. [Usar] • Demostrar la disposición de un array en la memoria y como se manipula punteros dentro de esos espacios de memoria. [Usar] • Demostrar el uso de aritmética de punteros y arrays.[Usar]
Lecturas : [Nak13], [stroustrup2013], [Reese2013], [Toppo2013]	

Unidad 3: Punteros y memoria dinámica (3)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Punteros y memoria dinámica - stack vs heap. • Alocación de memoria dinámica en C - malloc, calloc, realloc, free. • Punteros como retorno de una función en C/C++. • Punteros a funciones en C/C++. • Punteros a funciones y callback. • Memory leak en C/C++. 	<ul style="list-style-type: none"> • Mostrar la estructura de la memoria dentro de un programa y comprender como es que el compilador dispone elementos en el stack y en el heap.[Usar] • Demostrar el uso de las funciones y operadores de asignación de desasignación de memoria dinámica.[Usar] • Comprender las implicancias de retornar punteros desde funciones. [Usar] • Utilizar punteros a funciones como parámetros. [Usar] • Comprender la implicancia de uso de memoria dinámica y el memory leak. [Usar]
Lecturas : [Nak13], [stroustrup2013], [Reese2013], [Toppo2013]	

Unidad 4: Punteros y clases (5)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Punteros a miembros clase - atributos. • Punteros a miembros clase - métodos y llamadas a punteros a métodos. • Punteros a miembros clase - métodos static y llamadas a punteros a métodos static. • Punteros a clases - ejemplo con manejo de lista enlazada. 	<ul style="list-style-type: none"> • Comprender el uso de punteros a diferentes elementos de una clase. [Usar] • Comprender el uso de punteros a miembros estáticos de una clase. [Usar] • Introducir en la estructura nodo y su uso en una estructura de datos simple. [Usar] • Introducir a las estructura de datos, mostrando una implementación simple de listas enlazadas.[Usar]
Lecturas : [Nak13], [stroustrup2013], [Reese2013], [Toppo2013]	

Unidad 5: Functores (3)	
Resultados esperados: 1,3	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Definición de functores. • Functores y templates. • Paso de functores a funciones usando parámetros. • Paso de functores a funciones usando templates. • Paso de functores a clases usando parámetros. • Paso de functores a clases usando templates. • Ejemplos y aplicaciones. 	<ul style="list-style-type: none"> • Introducción a los functores. [Usar] • Uso de functores como parámetros a funciones y clases. [Usar] • Uso de functores en funciones y clases a través de templates. [Usar]
Lecturas : [Nak13], [stroustrup2013], [Reese2013], [Toppo2013]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

[Nak13] S. Nakariakov. *The Boost C++ Libraries: Generic Programming*. CreateSpace Independent Publishing Platforml, 2013.