

## 1. CURSO

CS113. Ciencia de la Computación II (Obligatorio)

## 2. INFORMACIÓN GENERAL

2.1 Curso	:	CS113. Ciencia de la Computación II
2.2 Semestre	:	3 <sup>er</sup> Semestre.
2.3 Créditos	:	4
2.4 horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Híbrido
2.8 Prerrequisitos	:	CS112. Ciencia de la Computación I. (2 <sup>do</sup> Sem) CS112. Ciencia de la Computación I. (2 <sup>do</sup> 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) S.O. Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (**Usar**)
- 3) S.O. Comunicarse efectivamente en diversos contextos profesionales. (**Usar**)
- 5) S.O. Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (**Usar**)

## 7. TEMAS

<b>Unidad 1: Introducción a Punteros en C/C++ (5)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje</b>
<ul style="list-style-type: none"> <li>• Declaración de punteros.</li> <li>• Trabajo con punteros: Referenciación. Desreferenciación.</li> <li>• Punteros tipados, aritmética de punteros, punteros void.</li> <li>• Punteros a punteros.</li> <li>• Punteros como argumentos de una función-llamada por referencia.</li> </ul>	<ul style="list-style-type: none"> <li>• Introducir en el manejo de punteros, sus operadores y su interacción en la memoria.[Usar]</li> <li>• Demostrar mediante ejemplos los diferentes usos de los operadores con punteros.[Usar]</li> <li>• Demostrar mediante ejemplos el uso de aritmética de punteros. [Usar]</li> <li>• Demostrar mediante ejemplos, las diferentes llamadas a funciones y el uso de punteros. [Usar]</li> </ul>
<b>Lecturas :</b> [Nak13], [Str13], [Ree13], [Top13]	

<b>Unidad 2: Manejo de Punteros con arrays (5)</b>	
<b>Resultados esperados: 1,3</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje</b>
<ul style="list-style-type: none"> <li>• Arrays como argumentos de una función.</li> <li>• Arrays de caracteres y punteros.</li> <li>• Punteros y Arrays de 2 dimensiones.</li> <li>• Punteros y arrays multidimensionales.</li> </ul>	<ul style="list-style-type: none"> <li>• Demostrar el uso de punteros con diferentes tipos de Arrays. [Usar]</li> <li>• Demostrar la disposición de un array en la memoria y como se manipula punteros dentro de esos espacios de memoria. [Usar]</li> <li>• Demostrar el uso de aritmética de punteros y arrays.[Usar]</li> </ul>
<b>Lecturas :</b> [Nak13], [Str13], [Ree13], [Top13]	

<b>Unidad 3: Punteros y memoria dinámica (3)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje</b>
<ul style="list-style-type: none"> <li>• Punteros y memoria dinámica - stack vs heap.</li> <li>• Alocación de memoria dinámica en C - malloc, calloc, realloc, free.</li> <li>• Punteros como retorno de una función en C/C++.</li> <li>• Punteros a funciones en C/C++.</li> <li>• Punteros a funciones y callback.</li> <li>• Memory leak en C/C++.</li> </ul>	<ul style="list-style-type: none"> <li>• 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]</li> <li>• Demostrar el uso de las funciones y operadores de asignación de desasignación de memoria dinámica.[Usar]</li> <li>• Comprender las implicancias de retornar punteros desde funciones. [Usar]</li> <li>• Utilizar punteros a funciones como parámetros. [Usar]</li> <li>• Comprender la implicancia de uso de memoria dinámica y el memory leak. [Usar]</li> </ul>
<b>Lecturas :</b> [Nak13], [Str13], [Ree13], [Top13]	

Unidad 4: Punteros y clases (5)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Punteros a miembros clase - atributos.</li> <li>• Punteros a miembros clase - métodos y llamadas a punteros a métodos.</li> <li>• Punteros a miembros clase - métodos static y llamadas a punteros a métodos static.</li> <li>• Punteros a clases - ejemplo con manejo de lista enlazada.</li> </ul>	<ul style="list-style-type: none"> <li>• Comprender el uso de punteros a diferentes elementos de una clase. [Usar]</li> <li>• Comprender el uso de punteros a miembros estáticos de una clase. [Usar]</li> <li>• Introducir en la estructura nodo y su uso en una estructura de datos simple. [Usar]</li> <li>• Introducir a las estructura de datos, mostrando una implementación simple de listas enlazadas.[Usar]</li> </ul>
Lecturas : [Nak13], [Str13], [Ree13], [Top13]	

Unidad 5: Functores (3)	
Resultados esperados: 1,3	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Definición de funtores.</li> <li>• Funtores y templates.</li> <li>• Paso de funtores a funciones usando parámetros.</li> <li>• Paso de funtores a funciones usando templates.</li> <li>• Paso de funtores a clases usando parámetros.</li> <li>• Paso de funtores a clases usando templates.</li> <li>• Ejemplos y aplicaciones.</li> </ul>	<ul style="list-style-type: none"> <li>• Introducción a los funtores. [Usar]</li> <li>• Uso de funtores como parámetros a funciones y clases. [Usar]</li> <li>• Uso de funtores en funciones y clases a través de templates. [Usar]</li> </ul>
Lecturas : [Nak13], [Str13], [Ree13], [Top13]	

## 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 Platform, 2013.
- [Ree13] M. Reese. *Understanding and Using C Pointers*. Inc., 2013.
- [Str13] B Stroustrup. *The C++ Programming Language, 4th edition*. Addison-Wesley, 2013.
- [Top13] H Toppo N. Dewan. *Pointers in C: A Hand on Approach*. Apress, 2013.