

Universidad Católica San Pablo (UCSP)

Escuela Profesional de Ciencia de la Computación

SILABO



CS113. Ciencia de la Computación II (Obligatorio)

1. Información general

1.1 Escuela	:	Ciencia de la Computación
1.2 Curso	:	CS113. Ciencia de la Computación II
1.3 Semestre	:	3 ^e Semestre.
1.4 Prerrequisitos	:	CS112. Ciencia de la Computación I. (2 ^{do} Sem)
1.5 Condición	:	Obligatorio
1.6 Modalidad de aprendizaje	:	Presencial
1.7 Horas	:	2 HT; 4 HP;
1.8 Créditos	:	4
1.9 Plan	:	Plan Curricular 2016

2. Profesores

Titular

- Alex Jesús Cuadros Vargas <acuadros@ucsp.edu.pe>
 - PosDoc en Ciencia de la Computación, ICMC-USP, Brasil, 2009.
 - Doctor en Ciencia de la Computación, ICMC-USP, Brasil, 2007.
 - Master en Ciencia de la Computación, ICMC-USP, Brasil, 2001.
- Christian Jorge Delgado Polar <cjdelgado@ucsp.edu.pe>
 - Master en Ciencia de la Computación, DCC-UFGM, Brasil, 2007.
- Gustavo Delgado Ugarte <ggdelgado@ucsp.edu.pe>
 - Máster en Ingeniería del Software, Escuela Universitaria de Ingeniería Industrial, Informática y Sistemas - UTA, Chile, 2009.

3. Fundamentación del 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.

4. Resumen

1. Introducción a Punteros en C/C++ 2. Manejo de Punteros con arrays 3. Punteros y memoria dinámica 4. Punteros y clases 5. Functores

5. Objetivos Generales

- 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. Contribución a los resultados (Outcomes)

Esta disciplina contribuye al logro de los siguientes resultados de la carrera:

- 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. Unidades

UNIDAD 1: Introducción a Punteros en C/C++

Resultados del estudiante: 1

Contenido	Objetivos Generales
<ul style="list-style-type: none">• Declaración de punteros.• Trabajo con punteros: 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: Nakariakov (2013), Stroustrup (2013), Reese (2013), Toppo (2013)	

UNIDAD 2 : Estructuras Algebraicas

Resultados del estudiante: 1

Contenido	Objetivos Generales
<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 cómo se manipulan punteros dentro de esos espacios de memoria. [Usar]• Demostrar el uso de aritmética de punteros y arrays. [Usar]
Lecturas: Nakariakov (2013), Stroustrup (2013), Reese (2013), Toppo (2013)	

UNIDAD 3: Punteros y memoria dinámica

Resultados del estudiante: 1

Contenido	Objetivos Generales
<ul style="list-style-type: none">• Punteros y memoria dinámica - stack vs heap.• Asignación 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.• Fuga de memoria en C/C++.	<ul style="list-style-type: none">• Mostrar la estructura de la memoria dentro de un programa y comprender cómo es que el compilador dispone elementos en el stack y en el heap. [Usar]• Demostrar el uso de las funciones y operadores de asignación y desasignación de memoria dinámica. [Usar]• Comprender las implicancias de retornar punteros desde funciones. [Usar]• Utilizar punteros a funciones como parámetros. [Usar]• Comprender la implicancia del uso de memoria dinámica y la fuga de memoria. [Usar]
Lecturas: Nakariakov (2013), Stroustrup (2013), Reese (2013), Toppo (2013)	

UNIDAD 4: Punteros y clases

Resultados del estudiante: 1

Contenido	Objetivos Generales
<ul style="list-style-type: none">• Punteros a miembros de clase - atributos.• Punteros a miembros de clase - métodos y llamadas a punteros a métodos.• Punteros a miembros de clase - métodos estáticos y llamadas a punteros a métodos estáticos.• 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 la estructura de nodo y su uso en una estructura de datos simple. [Usar]• Introducir las estructuras de datos, mostrando una implementación simple de listas enlazadas. [Usar]
Lecturas: Nakariakov (2013), Stroustrup (2013), Reese (2013), Toppo (2013)	

UNIDAD 5: Funtores	
Resultados del estudiante: 1	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> Definición de funtores. Funtores y plantillas. Paso de funtores a funciones utilizando parámetros. Paso de funtores a funciones utilizando plantillas. Paso de funtores a clases utilizando parámetros. Paso de funtores a clases utilizando plantillas. Ejemplos y aplicaciones. 	<ul style="list-style-type: none"> Introducción a los funtores. [Usar] Uso de funtores como parámetros en funciones y clases. [Usar] Uso de funtores en funciones y clases a través de plantillas. [Usar]
Lecturas: Nakariakov (2013), Stroustrup (2013), Reese (2013), Toppo (2013)	

8. Metodología

1. El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.
2. El profesor del curso presentará demostraciones para fundamentar clases teóricas.
3. El profesor y los alumnos realizarán prácticas.
4. Los alumnos deberán asistir a clase habiendo leído lo que el profesor va a presentar. De esta manera se facilitará la comprensión y los estudiantes estarán en mejores condiciones de hacer consultas en clase.

9. Evaluar

- **Evaluación Permanente 1:** 5%
- **Examen parcial:** 40%
- **Evaluación Permanente 2:** 5%
- **Examen final:** 50%

Referencias

- Nakariakov, S. (2013). *The Boost C++ Libraries: Generic Programming*. CreateSpace Independent Publishing Platforml.
- Reese, M. (2013). *Understanding and Using C Pointers*. Inc.
- Stroustrup, B (2013). *The C++ Programming Language, 4th edition*. Addison-Wesley.
- Toppo N. Dewan, H (2013). *Pointers in C: A Hand on Approach*. Apress.