

Universidad Católica San Pablo
Facultad de Ingeniería y Computación
Escuela Profesional de
Ciencia de la Computación
SILABO



CS102O. Objetos y Abstracción de Datos (Obligatorio)

2010-1

1. DATOS GENERALES

1.1 CARRERA PROFESIONAL	:	Ciencia de la Computación
1.2 ASIGNATURA	:	CS102O. Objetos y Abstracción de Datos
1.3 SEMESTRE ACADÉMICO	:	3 ^{er} Semestre.
1.4 PREREQUISITO(S)	:	CS101O. Introducción a la Programación Orientada a Objetos. (2 ^{do} Sem)
1.5 CARÁCTER	:	Obligatorio
1.6 HORAS	:	2 HT; 2 HP; 2 HL;
1.7 CRÉDITOS	:	4

2. DOCENTE

3. FUNDAMENTACIÓN DEL CURSO

Este es el tercer curso en la secuencia de los cursos introductorios a la informática. En este curso se pretende cubrir los conceptos señalados por la *Computing Curricula IEEE(c)-ACM 2001*, bajo el enfoque *functional-first*. El paradigma orientado a objetos nos permite combatir la complejidad haciendo modelos a partir de abstracciones de los elementos del problema y utilizando técnicas como encapsulamiento, modularidad, polimorfismo y herencia. El dominio de estos temas permitirá que los participantes puedan dar soluciones computacionales a problemas de diseño sencillos del mundo real.

4. SUMILLA

1. DS/Gráfos y Árboles.2. PF/Construcciones fundamentales.3. PF/Algoritmos y Resolución de Problemas.4. PF/Programación Orientada a Eventos.5. AL/Análisis Básico de Algoritmos.6. AL/Algoritmos Fundamentales.7. PL/Declaración y Tipos.8. PL/Mecanismos de Abstracción.9. PL/Programación Orientada a Objetos.10. SE/Diseño de Software.11. SE/Usando APIs.12. SE/Especificación de Requerimientos.

5. OBJETIVO GENERAL

- Introducir al alumno a los fundamentos del paradigma de orientación a objetos, permitiendo asimilar los conceptos necesarios para desarrollar un sistema de información.

6. CONTRIBUCIÓN A LA FORMACIÓN PROFESIONAL Y FORMACIÓN GENERAL

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

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. [Nivel Bloom: 3]
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. [Nivel Bloom: 3]
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. [Nivel Bloom: 3]
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. [Nivel Bloom: 3]
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. [Nivel Bloom: 3]
- k) Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. [Nivel Bloom: 3]

7. CONTENIDOS

UNIDAD 1: DS/Gráfos y Árboles.(7 horas)

Nivel Bloom: 3

OBJETIVO GENERAL

- Ilustrar con ejemplos la terminología básica de teoría de grafos y algunas de las propiedades y casos especiales de cada una.
- Mostrar diferentes métodos de recorrido en árboles y grafos.
- Modelar problemas en Ciencias de la Computación usando grafos y árboles.
- Relacionar grafos y árboles con estructura de datos, algoritmos y conteo.

CONTENIDO

- Árboles.
- Grafos no dirigidos.
- Grafos dirigidos.
- Árboles de expansión.
- Estrategias de recorrido.

Lecturas: [Nakariakov, 2013]

UNIDAD 2: PF/Construcciones fundamentales.(5 horas)

Nivel Bloom: 3

OBJETIVO GENERAL

- Analizar y explicar el comportamiento de programas simples involucrando las estructuras de programación fundamental cubiertas por esta unidad.
- Modificar y extender programas cortos que usan condicionales estándar, estructuras de control iterativas y funciones.
- Diseñar, implementar, probar y depurar un programa que use cada una de las siguientes estructuras fundamentales de programación: cálculos básicos, entrada y salida simple, estructuras estándar condicionales e iterativas y definición de funciones.
- Escoger la estructura apropiada condicional e iterativa para una estructura de programación dada.
- Aplicar técnicas de descomposición estructurada o funcional para dividir un programa en pequeñas partes.
- Describir los mecanismos de paso de parámetros.

CONTENIDO

- Sintaxis básica y semántica de un lenguaje de más alto nivel.
- Variables, tipos, expresiones y asignaciones.
- Entrada y salida simple.
- Estructuras de control condicionales e iterativas.
- Funciones y paso de parámetros.
- Descomposición estructurada.

Lecturas: [Stroustrup, 2013]

UNIDAD 3: PF/Algoritmos y Resolución de Problemas.(5 horas)	
Nivel Bloom: 4	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Discutir la importancia de los algoritmos en el proceso de solución de problemas. ▪ Identificar las propiedades necesarias de un buen algoritmo. ▪ Crear algoritmos para resolver problemas simples. ▪ Usar pseudocódigo o un lenguaje de programación para implementar, probar y depurar algoritmos para resolver problemas simples. ▪ Describir estrategias útiles para depuración. 	<ul style="list-style-type: none"> ▪ Estrategias para la solución de problemas. ▪ El rol de los algoritmos en el proceso de solución de problemas. ▪ Estrategias de implementación para algoritmos. ▪ Estrategias de depuración. ▪ El Concepto y propiedades de algoritmos.
Lecturas: [Stroustrup, 2013]	

UNIDAD 4: PF/Programación Orientada a Eventos.(2 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Explicar la diferencia entre programación orientada a eventos y programación por línea de comandos. ▪ Diseñar, codificar, probar y depurar programas de manejo de eventos simples que respondan a eventos del usuario. ▪ Desarrollar código que responda a las condiciones de excepción lanzadas durante la ejecución. 	<ul style="list-style-type: none"> ▪ Métodos para la manipulación de eventos. ▪ Propagación de eventos. ▪ Manejo de excepciones.
Lecturas: [Stroustrup, 2013]	

UNIDAD 5: AL/Análisis Básico de Algoritmos.(3 horas)	
Nivel Bloom: 2	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Determinar la complejidad de tiempo y espacio de algoritmos simples. 	<ul style="list-style-type: none"> ▪ Análisis asintótico de límites en los casos promedio y superior. ▪ Identificar la diferencias entre el comportamiento entre el mejor, mediano y peor caso.
Lecturas: [Stroustrup, 2013]	

UNIDAD 6: AL/Algoritmos Fundamentales.(3 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Implementar los algoritmos cuadráticos más comunes y los algoritmos de ordenamiento $O(N\log N)$. ▪ Diseñar e implementar una función de (<i>hash</i>) apropiada para una aplicación. ▪ Diseñar e implementar un algoritmo de resolución de colisiones para tablas de <i>hash</i>. ▪ Discutir la eficiencia computacional de los principales algoritmos de ordenamiento, búsqueda y (<i>hashing</i>). ▪ Discutir otros factores, además de la eficiencia computacional, que influyen en la elección de los algoritmos, tales como tiempo de programación, mantenimiento y el uso de patrones específicos de aplicación en los datos de entrada. ▪ Resolver problemas usando los algoritmos de grafos fundamentales, incluyendo búsqueda por amplitud y profundidad; caminos más cortos con uno y múltiples orígenes, cerradura transitiva, ordenamiento topológico y al menos un algoritmo de árbol de expansión mínima. ▪ Demostrar las siguientes capacidades: evaluar algoritmos, seleccionar una opción de un rango posible, proveer una justificación para tal elección e implementar el algoritmo.. 	<ul style="list-style-type: none"> ▪ Algoritmos numéricos simples. ▪ Búsqueda secuencial y binaria. ▪ Algoritmos cuadráticos de ordenamiento (selección, inserción). ▪ Algoritmos de tipo $O(N\log N)$ (Quicksort, heapsort, mergesort). ▪ Tablas de (<i>hash</i>) incluyendo estrategias de solución para las colisiones. ▪ Árboles de búsqueda binaria. ▪ Representación de grafos (Listas y Matrices de adyacencia). ▪ Recorridos por amplitud y profundidad. ▪ El algoritmo del camino más corto (algoritmos de Dijkstra y Floyd). ▪ Cerradura transitiva (algoritmo de Floyd). ▪ Árbol de expansión mínima (algoritmos de Kruskal y Prim). ▪ Ordenamiento Topológico.
Lecturas: [Stroustrup, 2013]	

UNIDAD 7: PL/Declaración y Tipos.(2 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Explicar el valor de los modelos de declaración, especialmente con respecto a la programación en mayor escala. ▪ Identificar y describir las propiedades de una variable, tales como su: dirección asociada, valor, ámbito, persistencia y tamaño. ▪ Discutir la incompatibilidad de tipos. ▪ Demostrar las diferentes formas de enlace, visibilidad, ámbito y manejo del tiempo de vida. ▪ Defender la importancia de los tipos y el chequeo de tipos para brindar abstracción y seguridad. ▪ Evaluar las ventajas y desventajas en el manejo del tiempo de vida (conteo por referencia vs. recolección de basura). 	<ul style="list-style-type: none"> ▪ La concepción de tipos como un conjunto de valores unidos a un conjunto de operaciones. ▪ Declaración de modelos (enlace, visibilidad, alcance y tiempo de vida). ▪ Vista general del chequeo de tipos. ▪ Recolección de basura.
Lecturas: [Stroustrup, 2013]	

UNIDAD 8: PL/Mecanismos de Abstracción.(5 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
	<ul style="list-style-type: none"> ▪ Procedimientos, funciones e iteradores como mecanismos de abstracción. ▪ Mecanismos de parametrización (referencia vs. valor). ▪ Registros de activación y administración de almacenamiento. ▪ Tipos de parámetros y tipos parametrizados. ▪ Módulos en lenguajes de programación.
Lecturas: [Stroustrup, 2013]	

UNIDAD 9: PL/Programación Orientada a Objetos.(7 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Justificar la filosofía de diseño orientado a objetos y los conceptos de encapsulación, abstracción, herencia y polimorfismo. ▪ Diseñar, implementar, probar y depurar programas simples en un lenguaje de programación orientado a objetos. ▪ Describir como los mecanismos de clases soportan encapsulación y ocultamiento de la información. ▪ Diseñar, implementar y probar la implementación de la relación es-un <i>IsKindOf</i> entre objetos usando jerarquía de clases y herencia. ▪ Comparar y contrastar las nociones de sobrecarga y sobreescritura de métodos en un lenguaje de programación. ▪ Explicar la relación entre la estructura estática de una clase y la estructura dinámica de las instancias de dicha clases. ▪ Describir como los iteradores acceden a los elementos de un contenedor. 	<ul style="list-style-type: none"> ▪ Diseño orientado a objetos. ▪ Encapsulación y ocultamiento de la información. ▪ Separación de comportamiento e implementación. ▪ Clases y subclasses. ▪ Herencia (sobreescritura, despacho dinámico). ▪ Polimorfismo (polimorfismo de subtipo vs. herencia). ▪ Jerarquías de clases. ▪ Clases de tipo colección y protocolos de iteración. ▪ Representaciones internas de objetos y tablas de métodos.
Lecturas: [Stroustrup, 2013]	
UNIDAD 10: SE/Diseño de Software.(5 horas)	
Nivel Bloom: 2	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Discutir las propiedades del buen diseño de software incluyendo la naturaleza y el rol de la documentación asociada. ▪ Conducir una revisión de diseño de software con material de código abierto utilizando lineamientos apropiados. 	<ul style="list-style-type: none"> ▪ Conceptos y principios fundamentales de diseño. ▪ El rol y uso de contratos. ▪ Patrones de diseño.
Lecturas: [Stroustrup, 2013]	

UNIDAD 11: SE/Usando APIs.(1 horas)	
Nivel Bloom: 2	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Explicar el valor de las interfaces para programación de aplicaciones (APIs) en el desarrollo de software. ▪ Usar navegadores de clases y herramientas relacionadas durante el desarrollo de aplicaciones usando APIs. ▪ Diseñar, implementar, probar y depurar programas que usan paquetes API de larga escala. 	<ul style="list-style-type: none"> ▪ Programación usando API. ▪ Diseño de API. ▪ Navegadores de clases (<i>Class browsers</i>) y herramientas relacionadas. ▪ Depuración en el entorno API. ▪ Introducción a la computación basada en componentes.
Lecturas: [Stroustrup, 2013]	

UNIDAD 12: SE/Especificación de Requerimientos.(1 horas)	
Nivel Bloom: 2	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Discutir los retos de mantener software heredado. ▪ Usar un método común, no formal para modelar y especificar (en la forma de un documento de especificación de requerimientos) los requerimientos para un sistema de software de tamaño medio. ▪ Traducir en lenguaje natural una especificación de requerimientos de software escrita en un lenguaje de especificación formal comunmente usado. 	<ul style="list-style-type: none"> ▪ Técnicas de modelamiento del análisis de requerimientos. ▪ Prototipeo. ▪ Conceptos básicos de técnicas de especificación formal.
Lecturas: [Stroustrup, 2013]	

8. METODOLOGÍA
<p>El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.</p> <p>El profesor del curso presentará demostraciones para fundamentar clases teóricas.</p> <p>El profesor y los alumnos realizarán prácticas</p> <p>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.</p>

9. EVALUACIONES
<p>Evaluación Permanente 1 : 20 %</p> <p>Examen Parcial : 30 %</p> <p>Evaluación Permanente 2 : 20 %</p> <p>Examen Final : 30 %</p>

Referencias

[Nakariakov, 2013] Nakariakov, S. (2013). *The Boost C++ Libraries: Generic Programming*. CreateSpace Independent Publishing Platform.

[Stroustrup, 2013] Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley, 4th edition.