

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



**CS101O. Introducción a la Programación Orientada a  
Objetos (Obligatorio)**

2010-1

**1. DATOS GENERALES**

1.1 CARRERA PROFESIONAL	:	Ciencia de la Computación
1.2 ASIGNATURA	:	CS101O. Introducción a la Programación Orientada a Objetos
1.3 SEMESTRE ACADÉMICO	:	2 <sup>do</sup> Semestre.
1.4 PREREQUISITO(S)	:	CS101F. Introducción a la Programación. (1 <sup>er</sup> Sem)
1.5 CARÁCTER	:	Obligatorio
1.6 HORAS	:	2 HT; 2 HP; 4 HL;
1.7 CRÉDITOS	:	5

**2. DOCENTE**

**3. FUNDAMENTACIÓN DEL CURSO**

Este es el segundo curso en la secuencia de los cursos introductorios a la informática.  
El curso servirá como puente entre el paradigma de la imperativo y el orientado al objeto, a demás introducirá a los participantes en los diversos temas del área de computación como: algoritmos, estructuras de datos, ingeniería del software, etc.

**4. SUMILLA**

1. PL/Visión General de los Lenguajes de Programación.2. PL/Máquinas Virtuales.3. PL/Declaración y Tipos.4. PF/Construcciones fundamentales.5. PL/Programación Orientada a Objetos.6. PF/Algoritmos y Resolución de Problemas.7. PF/Recursividad.8. AL/Análisis Básico de Algoritmos.9. AL/Algoritmos Fundamentales.

**5. OBJETIVO GENERAL**

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

**6. CONTRIBUCIÓN A LA FORMACIÓN PROFESIONAL Y FORMACIÓN GENERAL**

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

- Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. [Nivel Bloom: 3]
- Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. [Nivel Bloom: 4]
- Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. [Nivel Bloom: 3]
- Trabajar efectivamente en equipos para cumplir con un objetivo común. [Nivel Bloom: 3]
- Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. [Nivel Bloom: 3]
- Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. [Nivel Bloom: 3]

## 7. CONTENIDOS

### UNIDAD 1: PL/Visión General de los Lenguajes de Programación.(1 horas)

Nivel Bloom: 2

#### OBJETIVO GENERAL

- Listar la evolución de los lenguajes de programación identificando como es que su historia nos ha conducido a los paradigmas actuales.
- Identificar al menos una característica distintiva para cada uno de los paradigmas de programación cubiertos en esta unidad.
- Evaluar las ventajas y desventajas entre los diferentes paradigmas, considerando temas tales como: eficiencia de espacio, eficiencia en el tiempo (para ambas partes computadora y programador), seguridad y el poder de las expresiones.

#### CONTENIDO

- Breve revisión de los paradigmas de programación.
- Comparación entre programación funcional y programación imperativa
- Historia de los lenguajes de programación.

**Lecturas:** [Stroustrup, 2013], [Deitel, 2013]

### UNIDAD 2: PL/Máquinas Virtuales.(2 horas)

Nivel Bloom: 2

#### OBJETIVO GENERAL

- Describir la importancia y poder de la abstracción en el contexto de máquinas virtuales.
- Explicar los beneficios de los lenguajes intermedios en el proceso de compilación.
- Evaluar las ventajas y desventajas entre desempeño vs. portabilidad.

#### CONTENIDO

- El concepto de máquina virtual.
- Jerarquías de las máquinas virtuales.
- Lenguajes intermedios.

**Lecturas:** [Stroustrup, 2013], [Deitel, 2013]

### UNIDAD 3: PL/Declaración y Tipos.(2 horas)

Nivel Bloom: 3

#### OBJETIVO GENERAL

- 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.

#### CONTENIDO

- 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.

**Lecturas:** [Stroustrup, 2013], [Deitel, 2013]

<b>UNIDAD 4: PF/Construcciones fundamentales.(6 horas)</b>	
<b>Nivel Bloom: 5</b>	
<b>OBJETIVO GENERAL</b>	<b>CONTENIDO</b>
<ul style="list-style-type: none"> <li>▪ Analizar y explicar el comportamiento de programas simples involucrando las estructuras de programación fundamental cubiertas por esta unidad.</li> <li>▪ Modificar y extender programas cortos que usan condicionales estándar, estructuras de control iterativas y funciones.</li> <li>▪ 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.</li> <li>▪ Escoger la estructura apropiada condicional e iterativa para una estructura de programación dada.</li> <li>▪ Aplicar técnicas de descomposición estructurada o funcional para dividir un programa en pequeñas partes.</li> <li>▪ Describir los mecanismos de paso de parámetros.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Sintaxis básica y semántica de un lenguaje de más alto nivel.</li> <li>▪ Variables, tipos, expresiones y asignaciones.</li> <li>▪ Entrada y salida simple.</li> <li>▪ Estructuras de control condicionales e iterativas.</li> <li>▪ Funciones y paso de parámetros.</li> <li>▪ Descomposición estructurada.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013], [Deitel, 2013]	

<b>UNIDAD 5: PL/Programación Orientada a Objetos.(10 horas)</b>	
<b>Nivel Bloom: 3</b>	
<b>OBJETIVO GENERAL</b>	<b>CONTENIDO</b>
<ul style="list-style-type: none"> <li>▪ Justificar la filosofía de diseño orientado a objetos y los conceptos de encapsulación, abstracción, herencia y polimorfismo.</li> <li>▪ Diseñar, implementar, probar y depurar programas simples en un lenguaje de programación orientado a objetos.</li> <li>▪ Describir como los mecanismos de clases soportan encapsulación y ocultamiento de la información.</li> <li>▪ Diseñar, implementar y probar la implementación de la relación es-un <i>IsKindOf</i> entre objetos usando jerarquía de clases y herencia.</li> <li>▪ Comparar y contrastar las nociones de sobrecarga y sobreescritura de métodos en un lenguaje de programación.</li> <li>▪ Explicar la relación entre la estructura estática de una clase y la estructura dinámica de las instancias de dicha clases.</li> <li>▪ Describir como los iteradores acceden a los elementos de un contenedor.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Diseño orientado a objetos.</li> <li>▪ Encapsulación y ocultamiento de la información.</li> <li>▪ Separación de comportamiento e implementación.</li> <li>▪ Clases y subclases.</li> <li>▪ Herencia (sobreescritura, despacho dinámico).</li> <li>▪ Polimorfismo (polimorfismo de subtipo vs. herencia).</li> <li>▪ Jerarquías de clases.</li> <li>▪ Clases de tipo colección y protocolos de iteración.</li> <li>▪ Representaciones internas de objetos y tablas de métodos.</li> </ul>
<b>Lecturas:</b> [Smith, 2001], [Stroustrup, 2013], [Deitel, 2013]	

UNIDAD 6: PF/Algoritmos y Resolución de Problemas.(3 horas)	
Nivel Bloom: 4	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> <li>▪ Discutir la importancia de los algoritmos en el proceso de solución de problemas.</li> <li>▪ Identificar las propiedades necesarias de un buen algoritmo.</li> <li>▪ Crear algoritmos para resolver problemas simples.</li> <li>▪ Usar pseudocódigo o un lenguaje de programación para implementar, probar y depurar algoritmos para resolver problemas simples.</li> <li>▪ Describir estrategias útiles para depuración.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Estrategias para la solución de problemas.</li> <li>▪ El rol de los algoritmos en el proceso de solución de problemas.</li> <li>▪ Estrategias de implementación para algoritmos.</li> <li>▪ Estrategias de depuración.</li> <li>▪ El Concepto y propiedades de algoritmos.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013], [Deitel, 2013]	

UNIDAD 7: PF/Recursividad.(3 horas)	
Nivel Bloom: 4	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> <li>▪ Describir el concepto de recursividad y dar ejemplos de su uso.</li> <li>▪ Identificar el caso base y el caso general de un problema definido recursivamente.</li> <li>▪ Comparar soluciones iterativas y recursivas para problemas elementales tal como factorial.</li> <li>▪ Describir la técnica dividir y conquistar.</li> <li>▪ Implementar, probar y depurar funciones y procedimientos recursivos simples.</li> <li>▪ Describir como la recursividad puede ser implementada usando una pila.</li> <li>▪ Discutir problemas para los cuales el <i>backtracking</i> es una solución apropiada.</li> <li>▪ Determinar cuando una solución recursiva es apropiada para un problema.</li> </ul>	<ul style="list-style-type: none"> <li>▪ El concepto de recursividad.</li> <li>▪ Funciones matemáticas recursivas.</li> <li>▪ Funciones recursivas simples.</li> <li>▪ Estrategias de dividir y conquistar.</li> <li>▪ <i>Backtracking</i> recursivo.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013], [Deitel, 2013]	

UNIDAD 8: AL/Análisis Básico de Algoritmos.(2 horas)	
Nivel Bloom: 4	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> <li>▪ Determinar la complejidad de tiempo y espacio de algoritmos simples.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Identificar la diferencias entre el comportamiento entre el mejor, mediano y peor caso.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013], [Deitel, 2013]	

UNIDAD 9: AL/Algoritmos Fundamentales.(6 horas)	
Nivel Bloom: 4	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> <li>▪ Implementar los algoritmos cuadráticos más comunes y los algoritmos de ordenamiento <math>O(N\log N)</math>.</li> <li>▪ Diseñar e implementar una función de (<i>hash</i>) apropiada para una aplicación.</li> <li>▪ Diseñar e implementar un algoritmo de resolución de colisiones para tablas de <i>hash</i>.</li> <li>▪ Discutir la eficiencia computacional de los principales algoritmos de ordenamiento, búsqueda y (<i>hashing</i>).</li> <li>▪ 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.</li> <li>▪ 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.</li> <li>▪ Demostrar las siguientes capacidades: evaluar algoritmos, seleccionar una opción de un rango posible, proveer una justificación para tal elección e implementar el algoritmo..</li> </ul>	<ul style="list-style-type: none"> <li>▪ Algoritmos numéricos simples.</li> <li>▪ Búsqueda secuencial y binaria.</li> <li>▪ Algoritmos cuadráticos de ordenamiento (selección, inserción).</li> <li>▪ Algoritmos de tipo <math>O(N\log N)</math> (Quicksort, heapsort, mergesort).</li> <li>▪ Tablas de (<i>hash</i>) incluyendo estrategias de solución para las colisiones.</li> <li>▪ Árboles de búsqueda binaria.</li> <li>▪ Representación de grafos (Listas y Matrices de adyacencia).</li> <li>▪ Recorridos por amplitud y profundidad.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013], [Deitel, 2013]	

## 8. METODOLOGÍA

El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.

El profesor del curso presentará demostraciones para fundamentar clases teóricas.

El profesor y los alumnos realizarán prácticas

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. EVALUACIONES

**Evaluación Permanente 1** : 20 %

**Examen Parcial** : 30 %

**Evaluación Permanente 2** : 20 %

**Examen Final** : 30 %

## Referencias

[Deitel, 2013] Deitel, H. M. (2013). *C++ How to Program*. Pearson Educacion, 9th edition.

[Smith, 2001] Smith, J. A. (2001). *Desarrollo de Proyectos con Programación Orientada a Objetos con C++*. Thomson Learning.

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