

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



CS101F. Introducción a la Programación (Obligatorio)

2010-1

1. DATOS GENERALES

1.1 CARRERA PROFESIONAL	:	Ciencia de la Computación
1.2 ASIGNATURA	:	CS101F. Introducción a la Programación
1.3 SEMESTRE ACADÉMICO	:	1 ^{er} Semestre.
1.4 PREREQUISITO(S)	:	
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 primer 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-CS/ACM 2008*, bajo el enfoque *functional-first*.

La programación es uno de los pilares de la informática; cualquier profesional del área, necesitará programar para concretizar sus modelos y propuestas.

Este curso introducirá a los participantes en los conceptos fundamentales de este arte. Lo tópicos incluyen tipos de datos, estructuras de control, funciones, listas, recursividad y la mecánica de la ejecución, prueba y depuración.

El curso también ofrecerá una introducción al contexto histórico y social de la informática y una revisión del ámbito de esta disciplina.

4. SUMILLA

1. SP/Historia de la Computación.2. PL/Visión General de los Lenguajes de Programación.3. PL/Declaración y Tipos.4. PF/Construcciones fundamentales.5. PL/Programación Funcional.6. PF/Recursividad.7. AL/Algoritmos Fundamentales.8. PL/Mecanismos de Abstracción.9. PF/Algoritmos y Resolución de Problemas.10. PL/Máquinas Virtuales.11. PL/Programación Orientada a Objetos.12. SE/Usando APIs.

5. OBJETIVO GENERAL

- Introducir los conceptos fundamentales de programación y estructuras de datos utilizando un lenguaje funcional.
- Desarrollar su capacidad de abstracción, utilizar un lenguaje de programación funcional.

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]
- 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: 2]

7. CONTENIDOS

UNIDAD 1: SP/Historia de la Computación.(4 horas)

Nivel Bloom: 2

OBJETIVO GENERAL

CONTENIDO

- Listar las contribuciones de varios pioneros en el campo de la computación.
- Comparar la vida diaria antes y después del advenimiento de las computadoras personales e Internet.
- Identificar las tendencias continuamente significativas en la historia del campo de la computación.

- Prehistoria - el mundo antes de 1946.
- Historia del hardware de computadoras, software, redes.
- Pioneros de la computación.

Lecturas: [Brookshear, 2008], [Thompson, 2011], [Gutttag, 2013], [Zelle, 2010]

UNIDAD 2: PL/Visión General de los Lenguajes de Programación.(1 horas)

Nivel Bloom: 2

OBJETIVO GENERAL

CONTENIDO

- 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.

- Historia de los lenguajes de programación.
- Paradigmas de programación.

Lecturas: [Thompson, 2011], [Gutttag, 2013], [Zelle, 2010]

UNIDAD 3: PL/Declaración y Tipos.(1 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. ▪ Vista general del chequeo de tipos.
Lecturas: [Thompson, 2011], [Guttag, 2013], [Zelle, 2010]	

UNIDAD 4: PF/Construcciones fundamentales.(2 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ 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. 	<ul style="list-style-type: none"> ▪ 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: [Thompson, 2011], [Guttag, 2013], [Zelle, 2010]	

UNIDAD 5: PL/Programación Funcional.(1 horas)	
Nivel Bloom: 4	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Delinear las fortalezas y debilidades del paradigma de programación funcional. ▪ Diseñar, codificar, probar y depurar programas usando el paradigma funcional. ▪ Explicar el uso de funciones como datos, incluyendo el concepto de cerraduras. 	<ul style="list-style-type: none"> ▪ Panorama general y motivación de los lenguajes funcionales. ▪ Recursión sobre listas, números naturales, árboles y otros datos definidos recursivamente. ▪ Pragmáticas (depuración en dividir y vencerás, persistencia de las estructuras de datos).
Lecturas: [Thompson, 2011], [Guttag, 2013], [Zelle, 2010]	

UNIDAD 6: PF/Recursividad.(6 horas)	
Nivel Bloom: 4	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Describir el concepto de recursividad y dar ejemplos de su uso. ▪ Identificar el caso base y el caso general de un problema definido recursivamente. ▪ Comparar soluciones iterativas y recursivas para problemas elementales tal como factorial. ▪ Describir la técnica dividir y conquistar. ▪ Implementar, probar y depurar funciones y procedimientos recursivos simples. ▪ Describir como la recursividad puede ser implementada usando una pila. ▪ Determinar cuando una solución recursiva es apropiada para un problema. 	<ul style="list-style-type: none"> ▪ El concepto de recursividad. ▪ Funciones matemáticas recursivas. ▪ Funciones recursivas simples. ▪ Estrategias de dividir y conquistar.
Lecturas: [Thompson, 2011], [Guttag, 2013], [Zelle, 2010]	

UNIDAD 7: AL/Algoritmos Fundamentales.(4 horas)	
Nivel Bloom: 4	
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). ▪ Árboles de búsqueda binaria. ▪ Recorridos por amplitud y profundidad.
Lecturas: [Thompson, 2011], [Guttag, 2013], [Zelle, 2010]	

UNIDAD 8: PL/Mecanismos de Abstracción.(4 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Explicar como los mecanismos de abstracción soportan la creación de componentes de software reusables. ▪ Demostrar la diferencia entre paso de parámetros por valor y por referencia. ▪ Defender la importancia de la abstracción especialmente con respecto a la programación en mayor escala. 	<ul style="list-style-type: none"> ▪ Procedimientos, funciones e iteradores como mecanismos de abstracción. ▪ Mecanismos de parametrización (referencia vs. valor). ▪ Tipos de parámetros y tipos parametrizados. ▪ Módulos en lenguajes de programación.
Lecturas: [Thompson, 2011], [Guttag, 2013], [Zelle, 2010]	

UNIDAD 9: PF/Algoritmos y Resolución de Problemas.(10 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: [Thompson, 2011], [Guttag, 2013], [Zelle, 2010]	

UNIDAD 10: PL/Máquinas Virtuales.(1 horas)	
Nivel Bloom: 2	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Describir la importancia y poder de la abstracción en el contexto de máquinas virtuales. 	<ul style="list-style-type: none"> ▪ El concepto de máquina virtual.
Lecturas: [Thompson, 2011], [Guttag, 2013], [Zelle, 2010]	

UNIDAD 11: PL/Programación Orientada a Objetos.(4 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ 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 sobrescritura de métodos en un lenguaje de programación. 	<ul style="list-style-type: none"> ▪ Clases y subclases. ▪ Polimorfismo (polimorfismo de subtipo vs. herencia). ▪ Jerarquías de clases.
Lecturas: [Thompson, 2011], [Guttag, 2013], [Zelle, 2010]	

UNIDAD 12: SE/Usando APIs.(2 horas)	
Nivel Bloom: 3	
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. 	<ul style="list-style-type: none"> ▪ Programación usando API.
Lecturas: [Thompson, 2011], [Guttag, 2013], [Zelle, 2010]	

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

- [Brookshear, 2008] Brookshear, J. G. (2008). *Computer Science: An Overview*. Addison-Wesley, 10th edition. 0321524039.
- [Guttag, 2013] Guttag, J. V. (2013). *Introduction To Computation And Programming Using Python*. Mit Press, 2013 edition.
- [Thompson, 2011] Thompson, S. (2011). *The Craft of Functional Programming, 3E*. Addison Wesley.
- [Zelle, 2010] Zelle, J. (2010). *Python Programming: An Introduction to Computer Science*. Franklin, Beedle Associates Inc, 2nd edition.