

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
COORDINACIÓN DE FORMACIÓN BÁSICA
COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN
UNIVERSITARIA
DEPARTAMENTO DE ACTUALIZACIÓN CURRICULAR Y FORMACIÓN
DOCENTE

DESCRIPCIÓN GENÉRICA DE UNIDADES DE APRENDIZAJE

Descripción Genérica

Nombre: **Introducción a la Programación**

Etapas: **Básica**

Área de conocimiento: **Programación e ingeniería de software**

Plan 2008-1

Competencia:

Diseñar programas de computadora de propósito general que solucionen problemas del mundo real, mediante la integración de las técnicas que encierra los conceptos de programación. Mostrando una actitud crítica y de compromiso en la resolución de problemas, para promover la construcción de programas bien estructurados, documentados, eficientes, confiables y de fácil mantenimiento.

Evidencia de desempeño:

Exámenes teóricos, tareas extractase, reportes de prácticas de laboratorio, exposición oral y escrita del proyecto final.

Distribución	HC	HL	HT	HPC	HCL	HE	CR	Requisito
	2	2	2			2	8	Recomendada Diseño de algoritmos

Contenidos Temáticos

Unidad I. Introducción general

1. Introducción general
 - 1.1. Breve cronología de los paradigmas de la programación.
 - 1.2. Concepto de "programa"
 - 1.3. La edición, compilación y depuración de un programa
 - 1.4. Uso de las bibliotecas (librerías) disponibles (estándares) del lenguaje.

Unidad II. Uso de objetos: tipos, operadores, expresiones y sentencias

2. Clases, objetos y métodos.
 - 2.1. Constantes y variables.
 - 2.1.1. Tipos de números y los tipos carácter y cadena de caracteres ("string").
Definición de variables y constantes.
 - 2.1.2. Operaciones aritméticas, de concatenación de cadenas.
 - 2.1.3. Objetos, clases y métodos.
 - 2.1.4. Parámetros de los métodos y valores que devuelven.
 - 2.1.5. La construcción de objetos.
 - 2.1.6. Tipos booleanos y operadores lógicos.
 - 2.1.7. Operadores de relación.
 - 2.1.8. Operador de asignación y conversión implícita de tipos.

- 2.1.9. Métodos (funciones) de acceso que no modifican valores (estado del objeto) y modificadores.
- 2.1.10. Métodos (funciones) matemáticos
- 2.1.11. Métodos de para realizar entrada y salida de datos.
- 2.1.12. Referencias a objetos.
- 2.2. Expresiones y sentencias
 - 2.2.1. Expresiones aritméticas y lógicas
 - 2.2.2. Sentencias simples y compuestas (o complejas)
 - 2.2.3. Sentencia vacía ó nula
 - 2.2.4. Bloques de sentencias.

Unidad III. Implementar clases

- 3. Las clases como cajas negras: encapsulación
 - 3.1. Diseño de la interfase de una clase
 - 3.2. Definición de los métodos de la clase.
 - 3.3. Definición de los constructores.
 - 3.4. Campos de clase y campos de instancia.
 - 3.5. La documentación de una clase.
 - 3.6. La implementación de los métodos y los constructores.
 - 3.7. Las clases de variables:
 - 3.7.1. campos de instancia,
 - 3.7.2. variables locales,
 - 3.7.3. variables parámetro.
 - 3.8. Métodos estáticos.
 - 3.9. Prueba de una clase.

Unidad IV. Sentencias de condicionales y de iteración

- 4. El conocimiento básico para construir estructuras de control.
 - 4.1. Operaciones booleanas
 - 4.2. Valores booleanos, variables, operadores y expresiones
 - 4.3. Negación, conjunción, expresiones complejas, tautologías y equivalencia, leyes de De Morgan.
 - 4.4. Implicación
 - 4.5. Cálculo de predicados.

Unidad V. Estructuras de control

- 5. Las estructuras de control
 - 5.1. Ciclos
 - 5.2. Invariantes de ciclo.
 - 5.3. Corrección de los ciclos.
 - 5.4. Terminación de los ciclos.
 - 5.5. Sentencias condicionales: sus diversas variantes.
 - 5.6. Otras estructuras de control: multirramificación y la instrucción **goto** (si existe en el lenguaje usado)

Referencias bibliográficas actualizadas

Básica

Ceballos, Fco. Javier , Java 2 Curso de Programación, 3a Edición, Alfaomega

Ceballos, Fco. Javier , Microsoft C#. Lenguaje y Aplicaciones, Alfaomega Grupo Editor, ISBN 978-9701510940

Joshua Bloch, Effective Java Programming Language Guide (2nd Edition), Prentice

Hall PTR, ISBN 978-0321356680