

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA
COORDINACIÓN DE FORMACIÓN BÁSICA
PROGRAMA DE UNIDAD DE APRENDIZAJE HOMOLOGADA

1. Unidad académica (s): Facultad de Contaduría y Administración, Campus Tijuana
Facultad de Ciencias Administrativas, Campus Mexicali
Facultad de Ciencias Administrativas y Sociales, Campus Ensenada

2. Programa de Estudio: (Técnico, Licenciatura(s)): Licenciado en Informática 3. Vigencia del plan: 2009-2

4. Nombre de la Unidad de Aprendizaje: Programación Avanzada 5. Clave 11852

6. HC 2 HL 3 HT HPC HE 2 CR 7

7. Ciclo Escolar: 2011-1

8. Etapa de formación a la que pertenece Disciplinaria

9. Carácter de la Unidad de Aprendizaje: Obligatoria X Optativa

10. Requisitos para cursar la Unidad de Aprendizaje: Programación

UNIVERSIDAD AUTÓNOMA
DE BAJA CALIFORNIA



FAC. DE CIENCIAS
ADMINISTRATIVAS
MEXICALI

I. DATOS DE IDENTIFICACIÓN (Continuación)

Programa (s) de estudio: (Técnico, Licenciatura (s)) Lic. en Informática Vigencia del plan: 2009-2

Nombre de la unidad de aprendizaje: Programación Avanzada Clave: 11852

HC: 2 HL: 3 HT: HPC: HCL: HE: 2 CR: 7

Formuló:

LSC. Verónica Quizan García (Mxli)

MTRI. Adelaida Figueroa Villanueva (Mxli)

MTIC. Claudia Viviana Álvarez Vega (Mxli)

MC. Ray Brunett Parra Galaviz (Tijuana)

MC. José Manuel Valencia Moreno (Ensenada)

Dr. Eduardo Ahumada Tello (Tijuana)

MTRI. Jesús Antonio Padilla Sánchez (Ensenada)

M.C. Maria del Consuelo Salgado Soto (Tijuana)

Vo. Bo. M.P. Eva Olivia Martínez Lucero

Cargo: Subdirector FCA y S, Ensenada

Vo. Bo. M.A. Ernesto Alonso Pérez Maldonado

Cargo: Subdirector FCA, Mexicali

Vo. Bo. M. A. José Raúl Robles Cortez

Cargo: Subdirector FCA, Tijuana

Fecha:

1ro. de octubre del 2010

II. PROPÓSITO GENERAL DE LA UNIDAD DE APRENDIZAJE

Esta materia se encuentra en la etapa disciplinaria, de carácter obligatorio y pertenece al área de conocimiento de Programación e Ingeniería de Software y requiere haber cursado la unidad de aprendizaje de Programación, proporcionando a su vez conocimientos útiles en las unidades de aprendizaje para las unidades de aprendizaje de Desarrollo de Aplicaciones Web, Programación Empresarial.

Tiene como propósito, ofrecer al estudiante los conocimientos necesarios para desarrollar aplicaciones, utilizando el paradigma de Programación Orientada a Objetos y estructuras de datos dinámicas.

El contenido de esta unidad de aprendizaje asentará las bases para que el alumno desarrolle en su desempeño profesional aplicaciones bien fundamentadas.

III. COMPETENCIA (S) DE LA UNIDAD DE APRENDIZAJE

Generar aplicaciones computacionales, utilizando el paradigma de programación orientado a objetos y estructuras de datos dinámicas, para dar a solución de problemas específicos, con ética y creatividad

IV. EVIDENCIA (S) DE DESEMPEÑO

Desarrollar una aplicación computacional que resuelva un problema específico, que utilice los conceptos de Programación Orientada a Objetos, tales como Herencia, Polimorfismo, interacción con flujos de datos tanto de entrada como de salida y al menos una estructura de datos dinámica.

V. DESARROLLO POR UNIDADES

Competencia: Construir programas utilizando métodos y clases para estructurar aplicaciones apegadas al paradigma de programación orientada a objetos de manera eficaz y de forma responsable.

Duración 25 horas

Contenido

1. Métodos y Clases

1.1. Características de los métodos y clases

1.2. Métodos

1.2.1. Main

1.2.2. Constructores

1.2.3. Definidos por el programador

1.2.3.1. Devolución de valores

1.2.3.2. Visibilidad y accesibilidad

1.2.3.3. Paso de parámetros

1.2.3.4. Métodos abstractos

1.3. Clases

1.3.1. Predefinidas por el lenguaje.

1.3.2. Definidas por el programador

1.3.2.1. Clases simples y compuestas

1.3.2.1.1. Implementación de una clase

1.3.2.1.2. Acceso a miembros de una clase

1.3.2.1.3. Métodos de clases

1.3.2.1.4. Constructores

1.3.2.1.4.1. Constructores de inicialización por default

1.3.2.1.4.2. Sobrecarga de constructores

V. DESARROLLO POR UNIDADES

Competencia:

Diseñar programas reutilizando código para estructurar aplicaciones de manera eficiente con actitud deductiva y disciplina.

Duración 10 horas

Contenido

2. Herencia y Polimorfismo

2.1. Herencia

2.1.1. Herencia simple

2.1.1.1. Clase abstracta

2.1.1.2. Subclases

2.1.1.3. Superclases

2.1.2. Herencia múltiple

2.2. Polimorfismo

2.2.1. En métodos

2.2.1.1. Sobrecarga

2.2.1.2. Sobre-escritura

2.2.2. En variables

2.2.3. En Operadores

V. DESARROLLO POR UNIDADES

Competencia:

Crear programas utilizando flujos de archivos para manipular información almacenada con responsabilidad y honradez.

Contenido

Duración 5 horas

Unidad III. Archivos

3. Archivos

- 3.1. Flujos de entrada y salida
- 3.2. Clase Filtro
- 3.3. Clase File
- 3.4. Archivos secuenciales
- 3.5. Archivos de acceso aleatorio

V. DESARROLLO POR UNIDADES

Competencia:

Aplicar estructuras de manejo de memoria dinámica y estática identificando la estructura mas adecuada a la problemática planteada para manipular sus datos, de manera lógica y ordenada.

Contenido

Duración 40 horas

Unidad IV. Estructura de datos

4.1. Estructura dinámicas

4.1.1. Listas

4.1.2. Pilas

4.1.3. Cola

4.2. Recursividad

4.3. Arboles

4.3.1. Arboles binarios de búsqueda

4.3.2. Arboles binarios perfectamente equilibrados

4.4. Ordenación de datos

4.5. Búsqueda de datos

VI. ESTRUCTURA DE LAS PRÁCTICAS

No. de práctica	Competencia (s)	Descripción	Material de Apoyo	Duración
1	Construir programas utilizando métodos y clases para estructurar aplicaciones apegadas al paradigma de programación orientada a objetos de manera eficaz y de forma responsable.	Aplicar los conceptos de métodos y clases, mediante la creación de una clase que implemente sobre carga de constructores y contenga al menos 3 métodos, uno de ellos con paso de parámetros.	Computadora, entorno de desarrollo de software, lenguaje de programación y material bibliográfico.	3 Horas
2	Diseñar programas reutilizando código para estructurar aplicaciones funcionales de manera eficiente con actitud deductiva, con integridad y disciplina.	Comprobar el concepto de Herencia y Polimorfismo, desarrollando un programa que implemente dos niveles de herencia y una de las clases heredades tenga sobreescritura de métodos.	Computadora, entorno de desarrollo de software, lenguaje de programación y material bibliográfico.	3 Horas
3	Crear programas utilizando flujos de archivos para manipular información almacenada con responsabilidad y honradez.	Crear una clase con métodos que realicen operaciones de entrada y salida de información en archivos.	Computadora, entorno de desarrollo de software, lenguaje de programación y material bibliográfico.	6 Horas
4	Aplicar estructuras de manejo de memoria dinámica y estática identificando la estructura mas adecuada a la problemática planteada para manipular sus datos, de manera lógica y ordenada.	Construir una clase que simule el comportamiento de una estructura de datos dinámica, incluyendo un método que ordene los datos de la estructura.	Computadora, entorno de desarrollo de software, lenguaje de programación y material bibliográfico.	6 Horas

VII. METODOLOGÍA DE TRABAJO

1. Exposiciones por parte del profesor
2. Resolución de problemas en clase
3. Ejercicios extra clase
4. Consulta de tutoriales, manuales y libros

VIII. CRITERIOS DE EVALUACIÓN

1. Se realizarán tres evaluaciones parciales (30%)
2. Elaboración de prácticas en laboratorio (30%)
3. Desarrollo de un trabajo final (40%)

IX. BIBLIOGRAFÍA.

Básica

Joyanes, Luis; Zahonero, Ignacio. Programación en Java 2: Algoritmos, estructuras de datos y programación orientada a objetos. Mc Graw Hill, 1ra. Edición, 2002.

Ceballos, Francisco. Java 2 curso de programación. 3era edición, 2006. Alfaomega Ra-Ma.

Joyanes, Luis; Zahonero, Ignacio. Estructuras de datos en Java. Mc Graw Hill, 1ra. Edición, 2008.

Dean, John; Dean, Raymond. Introducción a la programación con Java. 1era. edición, 2009. Mc Graw Hill.

Cairo. Estructuras de Datos. 3era edición, 2006. Mc Graw Hill.

Complementaria

Sitio web
<http://sunsite.unam.mx/java.html>

Sitio web
<http://download.oracle.com/javase/7/docs/api/>

Wu, Thomas. Programación en Java: Introducción a la programación orientada a objetos. 2008. Mc Graw Hill