UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
COORDINACIÓN DE FORMACIÓN BÁSICA
COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN
PROGRAMA DE UNIDAD DE APRENDIZAJE POR COMPETENCIAS

I. DATOS DE IDENTIFICACIÓN

1. Unidad Académica: **Facultad de Ciencias**

2. Programa (s) de estudio: **Biólogo, Físico, Licenciado en Ciencias Computacionales, Matemático,**

3. Vigencia del plan: **2008-1**

4. Nombre de la Asignatura: **Introducción a la Programación** 5. Clave: 05

6. HC**: 2** HL **2** HT **2** HPC_____ HCL_____ HE __ CR **8**

7. Ciclo Escolar: **2008-1** 8. Etapa de formación a la que pertenece: **Básica**

9. Carácter de la Asignatura: Obligatoria _____**X**_____ Optativa _____

10. Requisitos para cursar la asignatura: Recomendada Diseño de Algoritmos

Formuló: **Francisco Juárez García y Oscar Mario Rodriguez** VoBo. M.C. Adrián Vázquez Osorio
Fecha: septiembre de 2007 Cargo: Subdirector

II. PROPÓSITO GENERAL DEL CURSO

La finalidad del curso de introducción a la programación es entrenar al alumno en el uso de los fundamentos de programación que le permitirán producir programas de calidad industrial. Estos fundamentos son:
La materia de Introducción a la Programación es obligatoria en la Licenciatura de Ciencias Computacionales y pertenece a la etapa Básica, se parte de que el alumno esta familiarizado con los conceptos básicos de programación dados en la materia de diseño de algoritmos. Las asignaturas subsecuentes directamente relacionadas con ésta son: Estructura de datos y Algoritmos y programación orienta a objetos.

III. COMPETENCIA (S) DEL CURSO

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.

IV. EVIDENCIA (S) DE DESEMPEÑO

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

V. DESARROLLO POR UNIDADES

Unidad I. Introducción general	**Competencia** Identificar los elementos que se involucran en un ambiente de programación, usando un ambiente integrado de desarrollo

Contenido Temático Duración: **2** horas

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**	**Competencia** Formular programas sencillos y correctos en los cuales se usen los conceptos de clase y objeto. y se utilicen la llamada a métodos (o funciones), los diferentes tipo de datos, operadores, expresiones y sentencias, que demuestren su familiaridad con los conceptos elementales que encierra el lenguaje de programación.

Contenido Temático Duración: **6** horas

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 aritméticos, 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 boléanos 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**	**Competencia** Diseñar programas que resuelvan situaciones problemáticas, y en los cuales se utilicen 2 o 3 clase diseñadas por el estudiante; para lo cual siguió un proceso de abstracción a fin de determinar las características esenciales de cada clase.

Contenido Temático Duración: **12** horas

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.

V. DESARROLLO POR UNIDADES

Unidad IV **Sentencias de condicionales y de iteración**	**Competencia** Diseñar programas que resuelvan situaciones problemáticas para las que se requiera utilizar sentencias condicionales, iterativas y ambas.

Contenido Temático Duración: **12 horas**

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, tautologias y equivalencia, leyes de De Morgan.
   - 4.4. Implicación
   - 4.5. Cálculo de predicados.
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)

V. DESARROLLO POR UNIDADES

Unidad V **Arreglos**	**Competencia** Diseñar programas que resuelvan situaciones problemáticas, y en los cuales resulte conveniente utilizar a los arreglos como estructuras que soportan la representación de los datos procesados por el programa

Contenido Temático Duración: **8** horas

5. El concepto de arreglo
   5.1. Declarar y usar arreglos
   5.2. Arreglos de objetos
   5.3. Arreglos bidimensionales
   5.4. Clases que manejan arreglos genéricos, específicas del lenguaje usado.
   5.5. Prueba y depuración

Unidad VI **Diseño de clases: acoplamiento y cohesión, precondiciones y poscondiciones**	**Competencia** El alumno criticará los programas que escriba en relación a los conceptos de cohesión, acoplamiento, precondiciones, poscondiciones e invariancia de clase.

Contenido Temático Duración: **18** horas

6. Los conceptos: Clases, objetos y relaciones.
   6.1. Identificar objetos y clases
   6.2. Cohesión y acoplamiento.
   6.3. Miembros estáticos de la clase
   6.4. Interfase de una clase.
   6.5. Los métodos consultores y los modificadores: cuando usarlos y cuando no.
   6.6. Precondiciones.
   6.7. Contratos: para depurar y para documentar.
   6.8. Poscondiciones.
   6.9. Invariantes de clase
   6.10. Prueba y depuración

Unidad VII **Herencia y polimorfismo**	**Competencia** Identificará las situaciones para las cuales sea conveniente utilizar la herencia y el polimorfismo.

Contenido Temático Duración: **12** horas

7. Los conceptos de herencia y polimorfismo en programación.
   7.1.La creación de subclases.
   7.2.La anulación o cambio de las características de métodos heredados.
   7.3.Jerarquías de clases
   7.4.Prueba y depuración

Unidad VIII **Excepciones**	**Competencia** Elegirá los mecanismos adecuados para que sus programas procesen correctamente las situaciones que pueden hacerlos fallar.

Contenido Temático Duración: **10** horas

8. El concepto de excepción.
   8.1. Errores y excepciones
   8.2. Excepciones no atrapadas
   8.3. Sentencias para el manejo de excepciones (try-catch en C# y Java, por ejemplo)
   8.4. Propagación de excepciones
   8.5. Prueba y depuración

Unidad IX **Recurrencia**	**Competencia** Identificará los problemas para los cuales resulte adecuado utilizar la recurrencia.
Contenido Temático 9. El concepto de recurrencia 9.1.Algoritmos expresados recursivamente. 9.2.La programación recursiva 9.3.Prueba y depuración	Duración: **8** horas

VI. ESTRUCTURA DE LAS PRÁCTICAS DE TALLER

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
1	Usar el editor y el depurador del ambiente de desarrollo seleccionado para el curso	Familiarizarse con diversos ambientes de compilación integrados y aprenda a compilar utilizando el compilador en línea. Se recomienda que el alumno tenga contacto con al menos dos plataformas.	Computadora, y acceso a Diferentes compiladores y diferentes sistemas operativos.	1 sesión
2	Familiarizar a los alumnos del curso en la manipulación de los tipos de datos, expresiones,	Presentar al alumno una serie de problemas de complejidad variable que le permitan adquirir un buen nivel de destreza en el manejo de las características elementales del lenguaje y su uso en la expresión de soluciones usando programas.	Computadora, compilador y problemas didácticos para ilustrar: tipos de datos y enunciados básicos.	2 sesiones
3			Computadora, compilador y problemas didácticos para ilustrar: el	sesiones

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
4			manejo de arreglos n-dimensionales, así como el uso de apuntadores Computadora, compilador y problemas didácticos para ilustrar: el manejo de estructuras con diversos niveles de complejidad	sesiones
5			Computadora y compilador	sesiones
6			Computadora y compilador	sesiones

VII. METODOLOGIA DE TRABAJO

Investigación Bibliográfica

La investigación bibliográfica será empleada en los trabajos extra-clase que se pedirán al estudiante sobre temas de actualidad o sobre temas que se verán posteriormente en clase. El propósito de estos trabajos es que el estudiante aprenda hacer investigación en medios electrónicos (Internet), libros, y revistas sobre temas del área. Las fuentes serán tanto en el idioma inglés como español. Los reportes deberán contener las referencias que se utilizaron para la realización del trabajo y debe contar imprescindiblemente una conclusión personal acerca de la investigación. El maestro debe enfatizar a los estudiantes que los reportes escritos sean claros y bien redactados, recalcándoles también las faltas de ortografía.

Aprendizaje participativo

Durante la clase de taller se aplicará está metodología en la que el estudiante juega un papel activo al intervenir en la planeación, realización y evolución del proceso de aprendizaje. Consiste básicamente en asignar un problema a cada equipo, el cual primeramente es analizado en forma individual, posteriormente en equipo, después se comentan las soluciones entre los diferentes equipos y al final se concluye. La participación del maestro en la aplicación de está metodología es de mediador.

Prácticas de laboratorio

Llevar a la práctica los conocimientos teóricos vistos en clase es el mejor método de enseñanza-aprendizaje, por eso es importante que el estudiante desarrolle habilidades con el manejo de diverso compiladores y sistemas operativos. Más aún se sugiere que se utilicen los problemas resueltos, a lápiz y papel, durante las sesiones de taller, para proveer retroalimentación al alumno de sus soluciones propuestas.

Exámenes de conocimientos

El maestro deberá aplicar al menos 2 exámenes de conocimientos durante el curso, de tal manera que refuercen los conocimientos aprendidos durante la clase. Los exámenes podrán ser de varios tipos, tales como: de preguntas abiertas, opción múltiple, crucigramas o mapas mentales.

VIII. CRITERIOS DE EVALUACION

La evaluación general del curso consistirá de exámenes teóricos, tareas-reportes, prácticas de laboratorio y una exposición oral con un reporte escrito.

Los porcentajes de evaluación serán los siguientes:

Exámenes	**40%**
Tareas/prácticas	30%
Proyecto final	30 %
Total	100%

Criterio de acreditación

- Resolver al menos 2 exámenes parciales en tiempo y forma.
- Las tareas se realizarán individualmente y las prácticas se realizarán por parejas o individualmente, según se indique.
- Cumplir con las prácticas y tareas extra-clase en tiempo y forma.
- Cumplir con la presentación del proyecto final en tiempo y forma.

Criterio de evaluación

- Las tareas, prácticas y exámenes serán resueltos en clase posterior de la entrega para que el estudiante conozca inmediatamente la solución propuesta en cada uno de los trabajos o exámenes.
- El proyecto final deberá realizarse en equipo de trabajo, la evaluación se dividirá en dos partes el 50% de la calificación será asignado al producto terminado y la segunda se asignará al un reporte escrito y a la exposición oral acerca del proyecto.

El reporte escrito será por equipo y los puntos a evaluar son, contenido, claridad y forma, así como ortografía y redacción; para la exposición oral los puntos a evaluar serán, dominio del tema, claridad y estructura. Los alumnos puede ayudarse en la exposición mediante apoyos visuales tales como proyector de transparencias, acetatos u otros medios.

IX. BIBLIOGRAFIA

Básica	Complementaria
Ceballos, Fco. Javier , Java 2 Curso de Programación, 3a Edicion, 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	