

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

I. DATOS DE IDENTIFICACIÓN

1. **Unidad Académica:** Facultad de Ingeniería, Mexicali; Facultad de Ingeniería y Negocios, Tecate; y Escuela de Ciencias de la Ingeniería y Tecnología, Valle de las Palmas.
2. **Programa Educativo:** Ingeniero en Mecatrónica
3. **Plan de Estudios:**
4. **Nombre de la Unidad de Aprendizaje:** Programación Orientada a Objetos
5. **Clave:**
6. **HC:** 02 **HL:** 02 **HT:** 01 **HPC:** 00 **HCL:** 00 **HE:** 02 **CR:** 07
7. **Etapas de Formación a la que Pertenece:** Básica
8. **Carácter de la Unidad de Aprendizaje:** Optativa
9. **Requisitos para Cursar la Unidad de Aprendizaje:** Ninguno



Equipo de diseño de PUA


José Manuel Villegas Izaguirre
Araceli Celina Justo López
Félix Francisco Reyna Beltrán

Firma


FRANCISCO REYNA

**Vo.Bo. de Subdirectores de
Unidades Académicas**

Alejandro Mungaray Moctezuma
Angélica Reyes Mendoza
María Cristina Castañón Bautista


MARÍA CRISTINA CASTAÑÓN B.

Firma



Fecha: 01 de junio de 2018

II. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

Esta unidad de aprendizaje tiene como propósito que el alumno conozca el paradigma de la programación orientada a objetos realizando aplicaciones en las que analice y diseñe modelos de clases, integre los conceptos de orientación a objetos y aplique la sintaxis de un lenguaje de programación logrando que éstas sean eficientes. Estos conocimientos le ayudarán en su razonamiento lógico para las unidades de aprendizaje de microcontroladores, automatización e instrumentación por computadora. En su campo laboral le facilitará el desarrollo de aplicaciones para la automatización e instrumentación por computadora.

Este programa de unidad de aprendizaje se encuentra en la etapa básica del plan de estudios y es de carácter optativo.

III. COMPETENCIA DE LA UNIDAD DE APRENDIZAJE

Desarrollar aplicaciones de software, a través de los principios de la orientación objetos, para solucionar problemas industriales que impliquen la automatización de procesos por computadora, de una forma analítica, responsable y honesta.

IV. EVIDENCIA(S) DE DESEMPEÑO

Desarrolla una aplicación de software que considere los principios de la orientación a objetos como son: abstracción, polimorfismo, herencia, encapsulamiento. Así mismo el software debe permitir el almacenamiento de datos en archivos.

V. DESARROLLO POR UNIDADES

UNIDAD I. Introducción a la programación orientada a objetos

Competencia:

Identificar y comprender el paradigma de la programación orientada a objetos, a través de los conocimientos básicos como el diseño de clases y objetos, para solución de problemas de software, con una actitud analítica, creativa y responsable.

Contenido:

Duración: 4 horas

- 1.1. Antecedentes
- 1.2. Características
 - 1.2.1. Encapsulación
 - 1.2.2. Polimorfismo
 - 1.2.3. Herencia
 - 1.2.4. Abstracción
 - 1.2.5. Cohesión
 - 1.2.6. Acoplamiento
- 1.3. Objetos
- 1.4. Clases
- 1.5. Relaciones entre clases
 - 1.5.1. Dependencia
 - 1.5.2. Generalización
 - 1.5.3. Asociación
 - 1.5.4. Agregación
 - 1.5.5. Composición
- 1.6. Métodos
- 1.7. Mensajes
- 1.8. Instancias

UNIDAD II. Introducción al lenguaje de programación orientada a objetos

Competencia:

Identificar la sintaxis del lenguaje de programación C++, aplicando sus elementos para construir aplicaciones, para brindar solución a los casos de estudio que se le presenten, siendo deductivos y perseverantes.

Contenido:

Duración: 4 horas

- 2.1. Características del lenguaje
- 2.2. Tipos de datos
 - 2.2.1. Enteros
 - 2.2.2. Reales/Flotantes
 - 2.2.3. Booleanos
 - 2.2.4. Cadenas
- 2.3. Operadores
- 2.4. Expresiones
- 2.5. Sentencias de selección
- 2.6. Estructuras de control de iteración
- 2.7 Arreglos

UNIDAD III. Elementos de la programación orientada a objetos

Competencia:

Utilizar los elementos del lenguaje de programación orientada a objetos, creando aplicaciones sencillas, para la solución de problemas computacionales, con una actitud analítica y responsable.

Contenido:**Duración: 8 horas**

- 3.1 Creación de clases y objetos
 - 3.1.1. Clases
 - 3.1.2. Atributos
 - 3.1.3. Métodos
 - 3.1.3.1. Paso de argumentos por referencia y por valor
 - 3.1.3.2. Métodos de acceso
 - 3.1.3.3. Sobrecarga /polimorfismo
 - 3.1.4. Modificadores de acceso
 - 3.1.5. Constructores
 - 3.1.6. Destructurres
 - 3.1.7. Arreglos de objetos

UNIDAD IV. Herencia

Competencia:

Comprender los conceptos de herencia, clase abstractas e interfaces, mediante la creación de aplicaciones de software, para la solución de problemas computacionales, con una actitud analítica, creativa y honesta.

Contenido:**Duración: 8 horas**

- 4.1. Concepto de herencia
- 4.2. Ventajas de herencia
- 4.3. Nomenclatura y reglas antecedentes de la programación
- 4.4. Clase Abstractas
- 4.5. Interfaces

UNIDAD V. Excepciones

Competencia:

Desarrollar aplicaciones de software considerando manejo de errores, mediante la implementación de excepciones, para solucionar problemas computacionales, de manera analítica y con responsabilidad.

Contenido:

Duración: 4 horas

- 5.1. Excepciones y errores
- 5.2. Clases de excepción
- 5.3. Tipos de excepciones
- 5.4. Bloque try... catch
- 5.5. Programación de una excepción

UNIDAD VI. Acceso al disco

Competencia:

Desarrollar aplicaciones de software, mediante el almacenamiento de información en el disco, para solucionar problemas computacionales, con una actitud propositiva, creativa y responsable.

Contenido:

Duración: 4 horas

- 6.1. Información sobre archivos y directorios o carpetas
- 6.2. Tipos de Archivos
- 6.3. Creación de carpetas
- 6.4. Clase para crear archivos
- 6.5. Escritura de Archivos
- 6.6. Lectura de Archivos

VI. ESTRUCTURA DE LAS PRÁCTICAS DE TALLER

No. de Práctica	Competencia	Descripción	Material de Apoyo	Duración
1	Analizar distintas clases y objetos cotidianos, identificando sus atributos y métodos, así como las relaciones entre ellos, para lograr un buen diseño orientado a objetos, con una actitud analítica, creativa y responsable.	Identifica clases, atributos y objetos del mundo real. Realiza modelado orientado a objetos reflejando relaciones entre clases. Redacta un reporte con la información identificada.	Presentaciones del docente, bibliografía sugerida y UML.	2 horas
2	Aplicar el lenguaje C++, para practicar su sintaxis del lenguaje, mediante la solución de problemas sencillos, con una actitud deductiva y perseverante.	Redacta un reporte con la solución propuesta.	Presentaciones del docente, bibliografía sugerida y compilador C++ para transcribir el código como actividad extraclase.	2 horas
3	Aplicar el lenguaje C++, utilizando los principios de la orientación a objetos, para la solución de problemas de automatización, con una actitud analítica y responsable.	Desarrolla ejercicios teóricos prácticos creando clases con atributos, métodos e instanciación de objetos. Desarrolla ejercicios teóricos prácticos utilizando la sobrecarga de funciones. Desarrolla ejercicios teóricos prácticos utilizando constructores y destructores. Desarrolla ejercicios teóricos prácticos creando arreglos de objetos. Redacta un reporte con la solución propuesta.	Presentaciones del docente, bibliografía sugerida y compilador C++ para transcribir el código como actividad extraclase.	4 horas
4	Aplicar herencia, clase abstractas e interfaces, mediante la resolución de ejercicios, para reutilizar código, con una actitud analítica, creativa y honesta.	Desarrolla ejercicios teórico práctico creando clases heredadas. Desarrolla ejercicios teórico práctico creando clases heredadas con arreglos de objetos.	Presentaciones del docente, bibliografía sugerida y compilador C++ para transcribir el código como actividad extraclase.	4 horas

		<p>Desarrolla ejercicios teóricos prácticos creando clases abstractas.</p> <p>Desarrolla ejercicios teóricos prácticos creando interfaces.</p>		
5	<p>Aplicar el manejo de excepciones, para atrapar y atender los errores que se puedan presentar en un programa en tiempo de ejecución, mediante la resolución de ejercicios, de manera analítica y con responsabilidad.</p>	<p>Desarrolla ejercicios teóricos prácticos creando excepciones.</p> <p>Desarrolla ejercicios teóricos prácticos lanzando excepciones.</p>	<p>Presentaciones del docente.</p> <p>Bibliografía sugerida.</p> <p>Compilador C++ para transcribir el código como actividad extraclase.</p>	2 horas
6	<p>Aplicar el almacenamiento y recuperación de información en el disco, para conservar los datos procesados, mediante la resolución de ejercicios, con una actitud propositiva, creativa y responsable.</p>	<p>Desarrolla ejercicios teóricos prácticos que guarden y consulte en archivos.</p> <p>Desarrolla ejercicios teóricos prácticos que editen y elimine en archivos.</p> <p>Redacta un reporte con la solución propuesta.</p>	<p>Presentaciones del docente.</p> <p>Bibliografía sugerida.</p> <p>Compilador C++ para transcribir el código como actividad extraclase.</p>	2 horas

VI. ESTRUCTURA DE LAS PRÁCTICAS DE LABORATORIO

No. de Práctica	Competencia	Descripción	Material de Apoyo	Duración
1	Identificar la forma de operar el laboratorio, los criterios de evaluación, derechos y obligaciones para la correcta operación del curso, mediante la revisión de documentos institucionales, con actitud positiva y con responsabilidad.	Se firman los acuerdos de evaluación, se lee el reglamento del laboratorio, se leen los derechos y obligaciones del estudiante de acuerdo al estatuto escolar.	Presentación del docente.	2 horas
2	Modelar distintas clases y objetos cotidianos, representando sus atributos y métodos, así como las relaciones entre ellos, para representar soluciones de problemas, con una actitud analítica, creativa y responsable.	Realiza modelado orientado a objetos reflejando relaciones entre clases. Entrega reporte del modelado realizado.	Documento Práctica de Laboratorio, bibliografía sugerida y compilador C++.	2 horas
3	Desarrollar aplicaciones en lenguaje C++, para practicar su sintaxis del lenguaje, mediante el desarrollo de programas, con una actitud deductiva y perseverante.	Desarrolla programas utilizando funciones de entrada y salida, operadores, expresiones y estructuras selectivas. Desarrolla programas utilizando arreglos y estructuras de iteración. Entrega el código realizado.	Documento Práctica de Laboratorio, bibliografía sugerida y compilador C++.	4 horas
4	Desarrollar aplicaciones en el lenguaje C++, utilizando los principios de la orientación a objetos, para la solución de problemas de automatización, mediante el desarrollo de programas, con una actitud analítica y responsable.	Desarrolla programas utilizando clases con atributos, métodos e instanciación de objetos. Desarrolla programas sobrecarga de funciones. Desarrolla programas utilizando constructores y destructores. Desarrolla programas utilizando arreglos de objetos.	Documento Práctica de Laboratorio, bibliografía sugerida y compilador C++.	8 horas
5	Desarrollar aplicaciones que utilicen	Desarrolla programas utilizando	Documento Práctica de	8 horas

	herencia, clase abstractas e interfaces, para reutilizar código, mediante el desarrollo de programas, con una actitud analítica, creativa y honesta	clases heredadas. Desarrolla programas utilizando clases heredadas con arreglos de objetos- Desarrolla programas utilizando clases abstractas. Desarrolla programas utilizando interfaces.	Laboratorio, bibliografía sugerida y compilador C++.	
6	Desarrollar aplicaciones que utilicen el manejo de excepciones, para atrapar y atender los errores que se puedan presentar en un programa en tiempo de ejecución, mediante el desarrollo de programas, de manera analítica y con responsabilidad.	Desarrolla programas utilizando excepciones. Desarrolla programas utilizando lanzando excepciones. Entrega el código realizado	Documento Práctica de Laboratorio, bibliografía sugerida y compilador C++.	4 horas
7	Desarrollar aplicaciones para el almacenamiento y recuperación de información en el disco, para mantener la información procesada, mediante el desarrollo de programas, con una actitud propositiva, creativa y responsable.	Desarrolla programas utilizando funciones para guardar y consultar en archivos. Desarrolla programas utilizando funciones para editar y eliminar en archivos.	Documento Práctica de Laboratorio, bibliografía sugerida y compilador C++.	4 horas

VII. MÉTODO DE TRABAJO

Encuadre: El primer día de clase el docente debe establecer la forma de trabajo, criterios de evaluación, calidad de los trabajos académicos, derechos y obligaciones docente-alumno.

Estrategia de enseñanza (docente)

- Guía el proceso enseñanza aprendizaje e introduce al estudiante en los contenidos del curso.
- Aplica el aprendizaje basado en problemas y ejercicios prácticos.
- Proporciona al alumno materiales bibliográficos para consulta.
- Propicia la participación activa del alumno en clase.
- Realiza ejercicios para ejemplificar las temáticas del curso.

Estrategia de aprendizaje (alumno)

- Resuelve problemas de ingeniería, por medio de la programación orientada a objetos.
- Participa en clase.
- Trabaja individual y en equipo.
- Realiza investigaciones documentales en diversas fuentes bibliográficas.

VIII. CRITERIOS DE EVALUACIÓN

La evaluación será llevada a cabo de forma permanente durante el desarrollo de la unidad de aprendizaje de la siguiente manera:

Criterios de acreditación

- 80% de asistencia para tener derecho a examen ordinario y 60% de asistencia para tener derecho a examen extraordinario de acuerdo al Estatuto Escolar artículos 70 y 71.
- Calificación en escala del 0 al 100, con un mínimo aprobatorio de 60.

Criterios de evaluación

- | | |
|---|------|
| - Tareas y trabajos en clase..... | 10% |
| - Evaluaciones parciales..... | 40% |
| - Evidencia de desempeño.....
(Aplicación de Software) | 50% |
| Total..... | 100% |

IX. REFERENCIAS

Básicas

- Ceballos, F. J. (2015). *C/C++ Curso de Programación* (4ª ed.). Editorial RA-MA.
- Ceballos, F. J. (2018). *Programación Orientada a Objetos con C++* (5ª ed.). Editorial RA-MA.
- Deitel, H. M. y Deitel, P. J. (2003). *Como programar en Java*. Pearson Educación. [clásica]
- Deitel, P.J. (2014). *Como Programar C++* (9ª ed.). Editorial Addison. Wesley.
- López, J.L. y Gutiérrez A. (2014). *Programación Orientada a Objetos C++ y Java. Un acercamiento interdisciplinario* (1ª ed.). Ebook. México. Editorial Patria. Recuperado: <https://www.editorialpatria.com.mx/pdf/files/9786074387711.pdf>
- Sierra, F. J. C. (2007). *Programación orientada a objetos con C++* (Vol. 3). Grupo Editorial RA-MA. [clásica]

Complementarias

- Fernandez, J. M. (2008). *Sun Certified Java Programmer*. [clásica]
- Jiménez, C. (2015). *UML: Aplicaciones en Java y C++*. Editorial RA-MA.
- Moldes, J. (2017). *JAVA 9 (Manual Imprescindible)*. Editorial Anaya Multimedia.

X. PERFIL DEL DOCENTE

El docente que imparta esta asignatura debe contar con título de Ingeniero en computación, sistemas computacionales o afín. Preferentemente con posgrado relacionado al área de programación; debe contar con experiencia docente deseable de dos años en asignaturas de programación o en el desarrollo de programas para la industria bajo plataformas como Java, C++, Python o similar. Poseer habilidades como facilidad para transmitir el conocimiento, ordenado, proactivo y responsable.