

PROGRAMA DE ASIGNATURA

I. IDENTIFICACIÓN DE LA ASIGNATURA.

| | | | | | |
|---|---|---------------------------------|--|-----|-------|
| Asignatura: Diseño y Programación Orientados a Objetos | | Sigla: ELO-329 | Fecha de aprobación 6 de marzo de 2017 | | |
| Créditos UTFSM: 3 | Prerrequisitos: IWI-131 o TEL-101 | Examen: No tiene | Unidad Académica que la imparte: | | |
| Créditos SCT: 5 | | | Departamento de Electrónica | | |
| Horas Cátedra Semanal: 3 | Ayudantía: No tiene | Laboratorio: No tiene | Semestre en que se dicta | | |
| | | | Impar X | Par | Ambos |
| Eje formativo: Ciencias de la Ingeniería Aplicada | | | | | |
| Tiempo total de dedicación a la asignatura: 145 horas | | | | | |

Descripción de la Asignatura

En esta asignatura los estudiantes aprenden a desarrollar soluciones de programación siguiendo el paradigma orientado a objetos. En su primera parte, la asignatura aborda conceptos de orientación a objetos al mismo tiempo que los aplica usando el lenguaje de programación Java. Este curso exige conocimientos básicos y práctica en algún lenguaje de programación como Python, C o Pascal. A través de Java se conoce y aplica el paradigma “programación dirigida por eventos”. Luego se estudian conceptos generales y herramientas usadas en ingeniería de software las cuales son aplicadas en el desarrollo de un proyecto grupal. En una tercera parte, se estudia el lenguaje de programación C++, destacando su enfoque orientado a objetos y su comparación con Java.

Requisitos de entrada

Utilizar lenguaje de programación estructurado como Python, C o Pascal.
Trabajar a nivel usuario en algún sistema operativo tipo Windows o Linux.

Contribución al perfil de egreso

1. Desarrollar servicios, productos, sistemas, y aplicaciones de las TIC, tanto en el contexto del desarrollo de oportunidades de innovación y emprendimientos tecnológicos, como para la optimización de procesos en las organizaciones y en la sociedad.
2. Desarrollar proyectos tecnológicos, con una visión del impacto global de las soluciones propuestas, considerando personas, recursos y tiempo, comprometido con la calidad, cumpliendo con el marco legal y ético, y atendiendo a la diversidad socio-cultural y el medio ambiente.
3. Comunicarse de manera efectiva, con un énfasis en el contexto técnico, de forma oral y escrita.

Resultados de Aprendizaje que se esperan lograr en esta asignatura.

1. Reconoce las características de los lenguajes orientados a objetos, definiendo cada una de ellas y reconociéndolas en contextos de uso. (1.1; 1.3; 1.4; 1.7)
2. Aplica el lenguaje Java en la construcción de sistemas de software de propósito general, extendiendo código ya existente como desarrollando aplicaciones a partir de nuevos requerimientos. (1.2; 1.3; 1.4; 1.5; 1.6; 1.7; 1.8; 1.9; 1.10)
3. Emplea el lenguaje C++ en la creación de soluciones de software de propósito general, adaptando código existente para satisfacer requerimientos nuevos o desarrollando

aplicaciones a partir de requerimientos nuevos. (3.1; 3.2; 3.3; 3.4; 3.5; 3.6; 3.7; 3.8; 3.9; 3.10)

4. Utiliza una metodología y algunas herramientas para el desarrollo de software, aplicando su metodología en un proyecto. (2.1; 2.2; 2.3)
5. Describe los niveles del estándar de certificación de software CMM, definiendo las características de cada uno y evaluando el nivel correspondiente a un proceso de desarrollo. (2.4)

Contenidos temáticos

1. Conceptos de orientación a objetos y Java.
 - 1.1. Conceptos en Lenguajes Orientados a Objetos: Objetos, clases, herencia, abstracción, polimorfismo, ligado dinámico, subtipo, y diseño orientado a objetos.
 - 1.2. Introducción a Java: Máquina virtual. Entornos de desarrollo para proyecto en Java. Tipos de datos y control de flujo básicos.
 - 1.3. Objetos, clases, herencia, clases abstractas, clase Object.
 - 1.4. Interfaces, clases anidadas: internas y anónimas.
 - 1.5. Programación basada en eventos (o dirigida por eventos): Conceptos.
 - 1.6. Programación de interfaces gráficas en Java. Swing.
 - 1.7. Manejo de excepciones.
 - 1.8. Generación de paquetes y documentación, Javadoc.
 - 1.9. Applets
 - 1.10. Administradores del Espacio (Layout Managers), Java Archives.
2. Ingeniería de software
 - 2.1. Introducción, Proceso de Desarrollo de Software.
 - 2.2. Metodologías de desarrollo de software: Desarrollo Iterativo e Incremental.
 - 2.3. Herramientas de apoyo al desarrollo de software: Casos de uso. Tarjetas CRC. UML.
 - 2.4. Niveles de Certificación CMM.
3. C++
 - 3.1. Clases en C++
 - 3.2. Constantes, referencias y cambios de tipo en C++
 - 3.3. Sobrecarga de Operadores
 - 3.4. Calificador Friend y Static
 - 3.5. Manejo de punteros y objetos en memoria dinámica
 - 3.6. Herencia en C++, Ligado dinámico en C++
 - 3.7. Excepciones
 - 3.8. Programación Genérica: Plantillas (templates) en C++, Objeto función (Functor).
 - 3.9. Biblioteca estándar de plantillas: Vector, Listas
 - 3.10. Algoritmos genéricos
 - 3.11. Patrones de Diseño

Metodología de enseñanza y aprendizaje.

1. Clases expositivas
2. Explicación y ejecución en clases de programas de ejemplo
3. Resolución grupal de tareas de programación
4. Desarrollo grupal de un proyecto
5. Presentaciones y demostraciones de estudiantes

Evaluación y calificación de la asignatura. (Ajustado a Reglamento Institucional- N°1)

| | |
|--|---|
| <p>Requisitos de aprobación y calificación</p> | <p>Proceso de evaluación y calificación: Se evalúa mediante 2 certámenes (C_1 y C_2), al menos 3 tareas (T) y un proyecto final (P_f).</p> <p>La nota promedio de certámenes se obtiene según:</p> $C_{Pr om} = \frac{C_1 + C_2}{2}$ <p>La nota promedio de tareas se calcula mediante el promedio simple de las notas obtenidas:</p> $T_{Pr om} = \frac{1}{n} \cdot \sum_{i=0}^n C_i$ <p>Nota de proyecto (NP): en ésta participan los estudiantes evaluando a sus pares.</p> <ul style="list-style-type: none"> • Promedio semestral (PS) se calcula según: <p>Si $C_{Pr om} \geq 50$:</p> $PS = C_{Pr om} * 0,55 + T_{Pr om} * 0,25 + NP * 0,2$ <p>Si $C_{Pr om} < 50$:</p> $PS = C_{Pr om} * 0,90 + T_{Pr om} * 0,06 + NP * 0,04$ <ul style="list-style-type: none"> • Los estudiantes que obtengan PS mayor o igual a 55 aprobarán la asignatura con nota final (NF): $NF = PS$ |
|--|---|

Recursos para el aprendizaje.

Bibliografía:

| | |
|---------------------------|--|
| Texto Guía | <ul style="list-style-type: none"> • Horstamm, C., & Cornell, G. (2013). <i>Core Java Volume I-- Fundamentals (9th Edition) (Core Series)</i>. USA: Prentice Hall. |
| Complementaria u Optional | <ul style="list-style-type: none"> • Horstamm, C., & Cornell, G. (2013). <i>Core Java Volume II-- Fundamentals (9th Edition) (Core Series)</i>. USA: Prentice Hall. • Lippman, S., Lajoie, J., Moo, B. (2015). <i>C++ Primer (5th Edition)</i> Massachusetts: Addison-Wesley • Plataforma virtual |

II. CÁLCULO DE CANTIDAD DE HORAS DE DEDICACIÓN- (SCT-Chile)- CUADRO RESUMEN DE LA ASIGNATURA.

| ACTIVIDAD | Cantidad de horas de dedicación | | |
|--|---------------------------------|---------------------|-------------------------|
| | Cantidad de horas por semana | Cantidad de semanas | Cantidad total de horas |
| PRESENCIAL | | | |
| Cátedra o Clases teóricas | 3 | 15 | 45 |
| Ayudantía/Ejercicios | | | |
| Visitas industriales (de Campo) | | | |
| Laboratorios / Taller | | | |
| Evaluaciones (certámenes, otros) | 1,5 | 2 | 3 |
| Otras (Presentación) | 3 | 1 | 3 |
| NO PRESENCIAL | | | |
| Ayudantía | | | |
| Tareas obligatorias | 5 | 8 | 40 |
| Estudio Personal (Individual o grupal) | 2 | 15 | 30 |
| Otras (Proyecto) | 6 | 4 | 24 |
| TOTAL (HORAS RELOJ) | | | 145 |
| Número total en CRÉDITOS TRANSFERIBLES | | | 5 |