**ANEXO I**

VIGENCIA DEL PROGRAMA:

|  |
| --- |
| ASIGNATURA: PROGRAMACIÓN CON OBJETOS I CÓD. DE MATERIA: 07033 |
| CARRERA: LICENCIATURA EN INFORMÁTICA CÓD. DE CARRERA: 288 |

**Ciclo Académico:** Inicial.

Año de la carrera: Primero.

Horas de clases semanales: 8 horas.

Teóricas: 3 horas.

Prácticas: 5 horas

Régimen de Cursado: Cuatrimestral / Segundo cuatrimestre.

Observaciones:

**ESPACIOS CURRICULARES CORRELATIVOS PRECEDENTES**

Aprobada/s: Algoritmos y Estructura de Datos (07030) Cursada/s y aprobada/s:

**ESPACIOS CURRICULARES CORRELATIVOS SUBSIGUIENTES**

Asignatura/s: Programación con Objetos II (07034)

**1-FUNDAMENTACIÓN**

Esta materia introduce al alumno en la programación orientada a objetos. Especialmente en los conceptos centrales del paradigma: objeto, mensaje y ambiente. Así como también las herramientas para la reutilización de código y modelado en dicho paradigma: clases, herencia, composición. Y herramientas de soporte como diagramas, documentación y testing. Resulta útil para la cursada el conocimiento previo de conceptos básicos de informática, en cuanto a arquitectura de computadoras, así como también fundamentos de programación (paradigma, programa, etc.). Articula con Algoritmos y Estructuras de Datos y Programación con Objetos II.

**2- OBJETIVOS GENERALES**

Construir modelos de problemas utilizando correctamente las nociones básicas del paradigma de objetos: objeto, mensaje, polimorfismo, estado interno, método, clase, herencia; aplicando en situaciones reales las nociones de estado y comportamiento del paradigma.

**3- OBJETIVOS ESPECÍFICOS**

Construir programas en un lenguaje de programación que soporte el paradigma de Objetos.

Aplicar los conocimientos adquiridos en la resolución de problemas concretos.

Utilizar apropiadamente las estructuras de datos básicas como pilas, colas, listas.

**4- CONTENIDO MÍNIMOS**

Programación Orientada a Objetos. Encapsulamiento y Abstracción. Herencia y Polimorfismo. Cohesión y Acoplamiento. Reutilización. Notación UML. Resolución de problemas y algoritmos. Tipo de dato abstracto. Estructuras de datos elementales: Pilas, Listas y Colas. Eventos, Excepciones. Persistencia de datos. Genericidad.

**5- PROGRAMA ANALÍTICO**

Unidad Nº 1: POO Conceptos Básicos.

Reseña histórica y motivos para el desarrollo de POO. Encapsulamiento y abstracción. Objetos y clases. Estructura interna de un Objeto. Asignación de responsabilidades de Objeto: métodos y mensajes. Ocultamiento de la información. Encapsular atributos y comportamiento. Notación UML.

Unidad Nº 2: Objetos y clases.

Clases e instancias. Creación de Objetos: métodos, constructores, clases y prototipos. Ciclo de vida de los objetos. Calificadores de acceso. Protección de estado. Lenguajes OO puros e híbridos. Costos y beneficios de uso POO. Concepto de modularidad aplicado a objetos.

Unidad Nº 3: Herencia

Definición y conceptos. Técnicas de herencia. Generalización-especialización. Jerarquía de clases. Extensión de clases. Herencia simple y múltiple. Clases abstractas y concretas. Análisis herencia vs. Composición. Reutilización.

Unidad Nº 4: Polimorfismo.

Introducción y concepto Polimorfismo. Código genérico y extensible. Binding dinámico y estático. Relaciones entre objetos. Importancia de un buen diseño de los objetos sobre la implementación.

Unidad Nº 5: Diseño por contratos.

Concepto de contratos en el diseño de objetos. Mecanismos de fiabilidad del software. Diseño de contratos y especificaciones. Principios de diseño de objetos. Cohesión y Acoplamiento.

Unidad Nº 6: Manejo de errores.

Gestión de errores. Ventajas del uso de excepciones. Detección de errores y excepciones. Conceptos generales. Lanzamiento y captura de una excepción. Sentencias de manejo de excepciones. Tipos de excepciones. Captura y delegación de excepciones. Programación genérica e introspección. API reflection.

**6- CRITERIOS DE EVALUACIÓN**

Los alumnos deben comprender los conceptos de programación orientada a objetos y demostrar su aplicación mediante la resolución de ejercicios prácticos de modelado e implementación.

**7- METODOLOGÍA DE TRABAJO PARA LA MODALIDAD PRESENCIAL**

1. Clases de teoría. En cada clase de teoría los alumnos conocerán de antemano los contenidos que se desarrollarán. Durante estas clases el profesor explicará los contenidos teóricos de la asignatura con abundante ejemplificación, que le permita luego al alumno abordar con solvencia la resolución de la guía de trabajos prácticos. Se fomentará la participación de los alumnos para lograr una dinámica que permita un mejor aprovechamiento de la clase.
2. Clases prácticas. Tienen los objetivos de: i) promover el estudio continúo de los alumnos y ii) de controlar la evolución del aprendizaje de los alumnos. En todas las clases prácticas los alumnos sabrán de antemano que problemas tienen que realizar. En ellas el profesor dará indicaciones para que los alumnos puedan superar las dificultades que les hayan aparecido en su resolución y resolverá individualmente las dudas y problemas que les hayan aparecido durante su resolución. Será labor del grupo de alumnos elaborar la colección de problemas resueltos de la asignatura.
3. Descripción de las Actividades Prácticas.

Formación Experimental: No se realizan actividades experimentales.

Resolución de Problemas del Mundo Real: Los alumnos deberán resolver Trabajos Prácticos que contienen problemáticas referidas a cada uno de los núcleos temáticos vistos, que corresponden a cada una de las Unidades de los contenidos analíticos. No son de entrega obligatoria.

Actividades de proyecto y diseño de sistemas informáticos: Trabajos Prácticos grupales que integran los conceptos vistos.

Instancias supervisadas de formación en la práctica profesional: No se realizan.

Otras actividades: Aprendizaje basado en problemas (uso de material extraído de las Olimpíadas Informáticas Argentina). Metodologías ágiles (programación extrema, programación de a pares)

1. Simulacros de parcial. La clase previa a cada evaluación parcial se llevará un simulacro que consistirá en plantear a los alumnos situaciones problemáticas similares a las que pueden evaluarse. Instándolos a resolverlos con el apoyo del material elaborado durante las clases, y luego haciendo una puesta en común con el docente y demás alumnos, lo que les permitirá hacer una valoración crítica ante la instancia parcial.

**8- ACREDITACIÓN DE ALUMNOS PRESENCIALES**

Regularización: Los alumnos deberán aprobar dos instancias de examen parcial, teniendo una instancia de recuperación donde pueden recuperar cualquiera de los dos parciales en el caso de haber desaprobado uno. Además, deberán realizar dos trabajos prácticos de diseño e implementación de un problema. Estos trabajos prácticos se van trabajando a lo largo de la cursada y el alumno tiene que demostrar suficiencia para resolverlo; el mismo cuenta con una fecha de entrega y pudiéndose recuperarse una vez.

Promoción: No aplica

**9- METODOLOGÍA DE TRABAJO PARA LOS ALUMNOS EN EL SISTEMA DE ASISTENCIA TÉCNICO PEDAGÓGICO**

Se realizarán a lo largo del cuatrimestre y previo a cada una de las instancias de evaluación clases de consulta para que los alumnos puedan comprobar su avance.

Los seguimientos de los trabajos prácticos se realizan de forma continua a lo largo del cuatrimestre, presentando soporte y ayuda a su resolución.

Los alumnos cuentan con una guía de ejercicios del mismo tenor de los necesarios para regularizar y aprobar la materia.

**10- ACREDITACIÓN DE ALUMNOS NO PRESENCIALES** (Modalidad a distancia)

Según Reglamento de Alumnos vigente.

**11- METODOLOGÍA DE TRABAJO SUGERIDA PARA EL APRENDIZAJE AUTOASISTIDO**.

Se recomienda la realización de los ejercicios de la guía de trabajos prácticos, así como también los Trabajos prácticos solicitados a los alumnos regulares. La bibliografía además cuenta con ejercitación y demostraciones del mismo tenor que las realizadas en clase. Se recomienda la asistencia al menos a dos clases de consultas previas al examen final libre.

**12- ACREDITACIÓN DE ALUMNOS LIBRES**

De acuerdo con el artículo 54 y 55 del Reglamento de Alumnos de la Universidad Nacional del Oeste, y otras reglamentaciones vigentes, se resuelve de la siguiente manera: Se tomará un examen teórico práctico. La parte práctica revisara sobre las ejercitaciones llevadas a cabo en el curso presencial. La parte teórica versará sobre la totalidad del contenido de la materia. La acreditación exige la aprobación de ambas partes.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **13- BIBLIOGRAFÍA** | | | | | | | | |
|  |  |  |  |  |  |  |  |  |
| **Apellido/s Nombre/s** | **Año Edición** | **Título de la Obra** | **Capítulo/ Tomo** | **Lugar de Edición** | **Editorial** | **Biblioteca** | **PMB** | **Otro** |
| **UNO** |
| Hortsmann, Cay S. | 2016, 10ma. | Core Java Volumen I y II. |  | New York | Pearson Prentice Hall |  |  |  |
| Deitel, Paul J. y Deitel, Harvey M. | 2012, 9na. | Java, cómo programar. |  |  | Pearson Prentice Hall |  |  |  |
| Sznajdleder, Pablo Augusto | 2012 | Algortimos a fondo. |  | Bs. As., Argentina | Alfaomega |  |  |  |
| Lewis, John y Chase Joseph | 2006 | Estructuras de datos con Java. |  | Madrid, España | Pearson Addison Wesley |  |  |  |
| Martin, Rober C. | 2004 | UML para Programadores Java. |  | Madrid, España | Pearson Prentice Hall |  |  |  |
| Ceballos Sierra, Francisco | 2011, 4ta. | Java 2 Curso de Programación. |  | México D. F., México | Alfaomega Grupo Editor |  |  |  |

|  |  |  |
| --- | --- | --- |
| **14- VIGENCIA DE PROGRAMA** | | |
| Año | Firma del Profesor Responsable | Aclaración Firma |
|  |  |  |
| Fecha | |

|  |  |  |
| --- | --- | --- |
| **15- VISADO ESCUELA** | | |
| Año | Firma Decano | Aclaración |
|  |  |  |
| Fecha: | |