



1. NOMBRE DE LA CARRERA: Licenciatura en Gestión de Tecnología

2. NOMBRE DE LA ASIGNATURA: Programación Avanzada I

Código de la asignatura: San Justo 1324 (Comisión 01-1900) – CABA 1324 (Comisión 02-5900)

Ciclo anual: 2024

Ciclo cuatrimestral: Primero

Modalidad de cursada:

3. CUERPO DOCENTE

Profesor a cargo de la asignatura: Ing. Gabriel Guzmán

Integrantes de la cátedra: Ing. Gabriel Guzmán

4. ASPECTOS ESPECÍFICOS

A. Carga horaria total: 64 hs

B. Carga horaria semanal: 4 hs

C. Carga horaria clases teóricas: 30 hs

D. Carga horaria práctica disciplinar: 34 hs

E. Carga horaria práctica profesional: N/A

F. Ubicación de la asignatura en el plan de estudios: 2do. Año – 1er. cuatrimestre

G. Correlatividades anteriores: N/A

H. Clases virtuales: SOLAMENTE los últimos lunes de cada mes se dictarán clases en modo virtual.

5. PROGRAMA

A. Fundamentación del marco referencial del programa.

Debido al avance continuo y vertiginoso de la tecnología y los consecuentes requerimientos de mayores niveles de especialización de los profesionales de sistemas, el dominio de disciplinas como la programación resulta ser un recurso indispensable para poder interpretar, aprehender y utilizar los nuevos conocimientos y desarrollos tecnológicos, para así asumir los cambios y participar en procesos de innovación y desarrollos vinculados a la tecnología

En este contexto, la Programación, constituye una herramienta tecnológica fundamental que permite la creación de modelos de hechos y / o situaciones de entidades diversa, desde aspectos



de la naturaleza, hasta el desarrollo de sistemas de gestión de información, pasando por la simulación de procesos hasta el diseño integral de aplicaciones.

Esta materia, además posee un buen grado de vinculación con otras materias del plan de estudios, como por ejemplo: Ingeniería de software, Investigación operativa, Arquitectura de software, Explotación y administración de BD.

Es destacable que esta asignatura sienta las bases para la materia consecutiva, Programación Avanzada II.

B. Contenidos previos.

Saber interpretar un diagrama de flujo, y conocer las estructuras básicas en algún lenguaje de programación.

C. Contenidos mínimos.

Programación en Java: sintaxis y semántica, tipos de datos, operadores, arreglos (arrays), estructuras de control de flujo, manejo de archivos, excepciones. Creación de interfaces gráficas para aplicaciones de escritorio. Paradigma de Programación Orientada a Objetos y Patrones de Diseño.

D. Objetivos generales.

Que el estudiante desarrolle profundos conocimientos de programación orientada a objetos, y de las buenas prácticas que hacen a este.

Que aprenda a diseñar interfaces gráficas y eventos programables.

E. Objetivos específicos.

Que adquiera habilidades en el desarrollo de aplicaciones, de escritorio, utilizando el lenguaje y tecnologías Java.;

Que aprenda a distinguir los diferentes patrones de diseño a aplicar para la resolución de determinados problemas de diseño.

Por otro lado, se espera que los alumnos muestren un alto grado de autosuficiencia en la búsqueda de soluciones a los problemas derivados de la codificación, es decir que puedan buscar y encontrar soluciones a sus problemas de código en páginas web especializadas y/o en la bibliografía provista por la cátedra. También se espera que puedan comprender textos básicos de programación y de patrones de diseño; siendo este último tópico, tema de investigación personal por parte del alumno.



F. Unidades didácticas.

Unidad N°1.

Introducción a Java

- Comprender las características propias del lenguaje JAVA.
- Entender la naturaleza de la máquina virtual.
- Abordar el estudio del lenguaje Java desde el desarrollo de aplicaciones utilizando el entorno de desarrollo integrado (IDE) Netbeans.

Contenido temático por unidad:

¿Qué es Java? Organización. La historia. El compilador. Máquina virtual. WORE(Write Once, Run Everywhere).Java Virtual Machine (JVM) . IDE a utilizar.

Bibliografía específica por unidad:

- Apuntes de Cátedra
- Deitel, H. M., & Deitel, P. J. (2010). Como programar en Java. Pearson Educación.
- Eckel, B. (2003). Thinking in JAVA. Prentice Hall Professional.
- Groussard, T. (2014). JAVA 8: Los fundamentos del lenguaje Java (con ejercicios prácticos corregidos). Ediciones Eni.
- Juárez, A. (2014). Java + programación orientada a objetos. Universidad de Buenos Aires.

Unidad N°2:

Sintaxis y Semántica de Java

- Familiarizarse con la sintaxis propia del lenguaje.
- Codificar las estructuras básicas de programación que posee JAVA.
- Entender la semántica del lenguaje.

Contenido temático por unidad:

¿Qué es una variable? Identificador. Definición de variables en Java. Tipos de dato primitivos. Carácter. Boolean. Enteros. Enteros Largos. Flotante. Operadores. ¿Qué es un Operador? Operadores Aritméticos. Operadores lógicos. Operadores de Asignación. Operador concatenación de caracteres. Operador condicional. Operadores incrementales y decrementales. Operadores relacionales. Operadores aplicables a bits. Clasificación. Sentencias. Bloques de Código. Expresiones. Caracteres especiales. Valores externos. Valores Constantes. Definición. Uso de constantes en Java. Comentarios. Tipos de Comentarios. Concepto de Referencia. Referencia this. Garbage Collection.

Bibliografía específica por unidad:

- Apuntes de Cátedra
- Deitel, H. M., & Deitel, P. J. (2010). Como programar en Java. Pearson Educación.
- Eckel, B. (2003). Thinking in JAVA. Prentice Hall Professional.



Universidad Nacional de La Matanza

ESCUELA DE FORMACIÓN CONTINUA

- Groussard, T. (2014). JAVA 8: Los fundamentos del lenguaje Java (con ejercicios prácticos corregidos). Ediciones Eni.
- Juárez, A. (2014). Java + programación orientada a objetos. Universidad de Buenos Aires.

Unidad N°3:

Estructuras de control de flujo

- Crear pequeños códigos funcionales.
- Entender la diferenciación de cada tipo de estructura de control.
- Utilizar sentencias condicionales anidadas.

Contenido temático por unidad:

Sentencias Condicionales. Bifurcación if. Bifurcación if else.
Bifurcación switch. Estructuras de Control (Bucles) . Bucle for.
Bucle while. Bucle do while Sentencia break.. Sentencia continue.

Bibliografía específica por unidad:

- Apuntes de Cátedra
- Deitel, H. M., & Deitel, P. J. (2010). Como programar en Java. Pearson Educación.
- Eckel, B. (2003). Thinking in JAVA. Prentice Hall Professional.
- Freeman, E., Robson, E., Bates, B., & Sierra, K. (2004). Head First Design Patterns: A Brain-Friendly Guide. " O'Reilly Media, Inc."
- Groussard, T. (2014). JAVA 8: Los fundamentos del lenguaje Java (con ejercicios prácticos corregidos). Ediciones Eni.
- Juárez, A. (2014). Java + programación orientada a objetos. Universidad de Buenos Aires.
- Rumbaugh, J., Booch, G., & Jacobson, I. (2000). El lenguaje unificado de modelado UML. Addison-Wesley.
- Weitzenfeld, A. (2005). Ingeniería de software orientada a objetos con UML, Java e Internet. Thomson

Unidad N°4:

Excepciones

- Diseñar y construir Excepciones.
- Definir Errores
- Comprender el flujo de las excepciones que provee JAVA

Contenido temático por unidad:

Definición Bloques try, catch y finally . Tipos de excepciones. Definición de errores.

Bibliografía específica por unidad:

- Apuntes de Cátedra
- Deitel, H. M., & Deitel, P. J. (2010). Como programar en Java. Pearson Educación.
- Eckel, B. (2003). Thinking in JAVA. Prentice Hall Professional.



Universidad Nacional de La Matanza

ESCUELA DE FORMACIÓN CONTINUA

- Groussard, T. (2014). JAVA 8: Los fundamentos del lenguaje Java (con ejercicios prácticos corregidos). Ediciones Eni.
- Juárez, A. (2014). Java + programación orientada a objetos. Universidad de Buenos Aires.

Unidad N°5:

Vectores, Matrices y Manejo de archivos

- Comprender la estructura interna de vectores y matrices.
- Manipular archivos de texto.
- Entender los distintos tipos de persistencia soportados por java.

Contenido temático por unidad:

Introducción. Creación. Inicialización en la declaración. Recorrido del array. Copia de arrays. Archivos texto, su creación, modos de acceso, posicionamiento, cierre, eliminación, y funciones relacionadas.

Bibliografía específica por unidad:

- Apuntes de Cátedra
- Deitel, H. M., & Deitel, P. J. (2010). Como programar en Java. Pearson Educación.
- Eckel, B. (2003). Thinking in JAVA. Prentice Hall Professional.
- Groussard, T. (2014). JAVA 8: Los fundamentos del lenguaje Java (con ejercicios prácticos corregidos). Ediciones Eni.
- Juárez, A. (2014). Java + programación orientada a objetos. Universidad de Buenos Aires.

Unidad N°6:

Interfaz Gráfica de Usuario

- Armar una interfaz gráfica con menús y ventanas.
- Familiarizarse con los contenedores y componentes gráficos que facilita el IDE.
- Introducir a los alumnos a la programación de eventos.

Contenido temático por unidad:

Descripción. Contenedores gráficos. Componentes gráficos. Eventos. Utilización de la estructura de control IF. Utilización de la estructura de control WHILE. Utilización de la estructura de control FOR. Utilización de Vectores. Introducción a Interfaz Gráfica de Usuario. Construcción de la Interfaz Gráfica.. Tópicos de Programación de Eventos.

Bibliografía específica por unidad:

- Apuntes de Cátedra
- Carrillo, M. G., & Herranz, M. D. S. M. (2005). Interfaces gráficas en Java. Editorial Universitaria Ramón Areces.

Unidad N°7:

Introducción a Programación Orientada a Objetos

- Comprender el modelado de objetos.
- Detectar clases, atributos y operaciones.



- Utilizar constructores.
- Implementar relaciones simples y múltiples.
- Trabajar con Herencia y Polimorfismo.

Contenido temático por unidad:

¿Qué es una clase? ¿Qué es un objeto? ¿Qué son los atributos? Atributos de instancia. Atributos de clase. ¿Qué son los métodos? Métodos de instancia. Métodos de clase. Encapsulamiento. Definición. Métodos de acceso. Constructores y destructores. El constructor. El destructor. Herencia. ¿Qué es el casting? Casteo implícito (widening casting). Casteo explícito (narrowing casting). Upcasting. Polimorfismo. Definición. Con redefinición. Sin redefinición. Clase abstracta. Interfáz. ¿Qué es una interfaz? Paquetes. Que es un paquete. La keyword final. Aplicable a atributos. Aplicable a métodos. Aplicable a clases.

Bibliografía específica por unidad:

- Apuntes de Cátedra
- Deitel, H. M., & Deitel, P. J. (2010). Como programar en Java. Pearson Educación.
- Eckel, B. (2003). Thinking in JAVA. Prentice Hall Professional.
- Groussard, T. (2014). JAVA 8: Los fundamentos del lenguaje Java (con ejercicios prácticos corregidos). Ediciones Eni.
- Juárez, A. (2014). Java + programacion orientada a objetos. Universidad de Buenos Aires.

Unidad N°8:

Introducción a los Patrones de diseño

- Entender qué son los patrones de diseño y cuándo utilizarlos
- Identificar y trabajar con los distintos tipos de patrones
- Comprender qué problema resuelve y cómo se implementa la solución a través de código de programación, ya sea en tecnologías.
- Reconocer los anti-patrones, junto con su propósito y utilización
- Analizar un problema determinado y decidir qué patrón de diseño utilizar para resolverlo
- Codificar la solución de un problema en un lenguaje de programación.
- Comprender bibliografía específica acerca de los patrones de diseño.

Contenido temático por unidad:

Patrón: concepto. Patrones de diseño: concepto. Historia. ¿Cuándo utilizarlos? ¿Cuándo no utilizarlos? Donde utilizarlos. Que son los patrones gof. Beneficios. Clasificación según su propósito. Creacionales. De comportamiento. Estructurales. 8



Bibliografía específica por unidad:

- Freeman, E., Robson, E., Bates, B., & Sierra, K. (2004). Head First Design Patterns: A Brain-Friendly Guide. " O'Reilly Media, Inc."
- Larman, C. (1999). UML y Patrones. Introducción al análisis y diseño orientado a objetos. Ed.
- Rumbaugh, J., Booch, G., & Jacobson, I. (2000). El lenguaje unificado de modelado UML. Addison-Wesley.

G. Bibliografía general.

- Carrillo, M. G., & Herranz, M. D. S. M. (2005). Interfaces gráficas en Java. Editorial Universitaria Ramón Areces.
- Deitel, H. M., & Deitel, P. J. (2010). Como programar en Java. Pearson Educación.
- Eckel, B. (2003). Thinking in JAVA. Prentice Hall Professional.
- Freeman, E., Robson, E., Bates, B., & Sierra, K. (2004). Head First Design Patterns: A Brain-Friendly Guide. " O'Reilly Media, Inc."
- Groussard, T. (2014). JAVA 8: Los fundamentos del lenguaje Java (con ejercicios prácticos corregidos). Ediciones Eni.
- Juárez, A. (2014). Java + programación orientada a objetos. Universidad de Buenos Aires.
- Larman, C. (1999). UML y Patrones. Introducción al análisis y diseño orientado a objetos. Ed.
- Rumbaugh, J., Booch, G., & Jacobson, I. (2000). El lenguaje unificado de modelado UML. Addison-Wesley.

6. METODOLOGÍA

Las clases se dictarán en el laboratorio, abordando y exponiendo cada clase con el uso del cañón, invitando a los alumnos a seguir lo enseñado en tiempo real en las máquinas.

Pueden traer sus notebooks.

7. MECANISMOS DE SEGUIMIENTO, SUPERVISIÓN Y EVALUACIÓN DE LAS ACTIVIDADES, PRESENCIALES Y/O DE SEGUIMIENTO VIRTUAL

Régimen de parciales

Los cursantes deberán rendir dos parciales durante el cuatrimestre. Podrán recuperar uno de los mismos (1° o 2°) al final del cuatrimestre aquellos que no alcancen la nota mínima para promover en alguna de las evaluaciones. Los parciales para ser aprobados deberán tener una nota no menor de siete (7), correspondiéndole cuando el estudiante conteste en forma apropiada por lo menos el 70% de los contenidos evaluados y previamente enunciados a criterio de la cátedra. En caso de rendir un examen recuperatorio la nota obtenida sustituirá la anterior. Resultarán promocionados los estudiantes de obtengan siete (7) o más en cada una de las evaluaciones o sus recuperatorios. Los estudiantes que obtengan notas entre cuatro (4) y menos de siete (7) quedaran en situación de cursada debiendo dar un examen final. En todos los casos deberán haber cumplido satisfactoriamente con los objetivos propuestos en los trabajos prácticos. Los que tengan menos de cuatro (4) quedarán en condición de insuficiente y deberán recurrar la asignatura. Para



los exámenes finales deberán cumplimentarse idénticos requisitos a los de la cursada los que serán evaluados de forma escrita y oral en fechas especiales que determine el departamento. La nota final se obtendrá como promedio de la totalidad de las notas obtenidas.

Trabajos Prácticos

Se formarán comisiones de trabajo a libre elección de los estudiantes de no más de 5 integrantes.

La materia tendrá varios trabajos prácticos orientativos y un trabajo final obligatorio, el cual consistirá en el desarrollo de una aplicación de escritorio programada en JAVA, la cual deberá incluir todos los contenidos explicados en la materia.

El trabajo práctico, así como su defensa son requisitos indispensables para la aprobación de la asignatura, promediándose la nota derivada del trabajo y su defensa con el segundo parcial.



8. CRONOGRAMA DE CLASES Y EXÁMENES

Importante: Los últimos lunes de cada mes se dictarán las clases en modalidad virtual.

Contenidos/ Actividades/ Evaluaciones	Semanas															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Nº de unidades / Parciales																
1. Introducción a Java																
2. Sintaxis y Semántica de Java																
3. Estructuras de control de flujo																
4. Excepciones																
5. Vectores, Matrices y Manejo de Archivos																
Prácticas en laboratorios – Entregas de TP – Repaso																
1º Parcial																
6. Interfaz Gráfica de Usuario																
7. Introducción a la Programación Orientada a Objetos																
8. Introducción a UML																
9. Introducción a los Patrones de diseño																
Prácticas en laboratorios – Entregas de TP – Repaso																
2º Parcial																
Recuperatorio – Cierre de notas																



9. CONDICIONES GENERALES PARA LA APROBACIÓN DE LA ASIGNATURA

A. Asistencia: Se requiere una asistencia a clases no menor al setenta y cinco (75%) sobre el total de la carga horaria de la asignatura.

B. Evaluación:

Se disponen de cuatro estados académicos posibles:

- + Ausente: cuando el alumno no tenga calificación en alguno de sus exámenes (o su recuperatorio).
- + Reprobada: cuando el alumno obtenga como calificación final de 1 a 3 puntos.
- + Cursada: cuando el alumno obtenga entre 4 y 6 puntos como calificación final.
- + Promocionada: cuando el alumno obtenga como calificación final entre 7 y 10 puntos.

Para las asignaturas cuatrimestrales habrá 2 instancias parciales y la posibilidad de 1 instancia recuperatoria. La calificación obtenida en el examen recuperatorio reemplaza y anula a todos los efectos, la obtenida en el examen parcial que se recupera.

A los fines de conformar la calificación final, los parciales no se promedian, salvo que ambas evaluaciones sean reprobadas, o ambas cursadas, o ambas promocionadas.

El alumno que culmine la materia en condición "cursada", deberá aprobar el examen final para tener la asignatura como aprobada.

Fecha de presentación: 09/02/2024

X *Gabriel E. Guzmán*

Ing. Gabriel E. Guzmán
Docente TITULAR de CÁTEDRA