



INSTITUTO/S: Tecnología e ingeniería

CARRERA/S: Licenciatura en informática / Tecnicatura universitaria en informática

MATERIA: Programación con objetos I

NOMBRE DEL RESPONSABLE DE LA ASIGNATURA: Gonzalez Tulian, Gerardo Martin

EQUIPO DOCENTE: Spigariol Lucas, Pettinari Joaquin

CUATRIMESTRE: 2^{do} **AÑO:** 1^{ro}

PROGRAMA N°:

Instituto/s: Tecnología e ingeniería

Carrera/s: Licenciatura en informática / Tecnicatura universitaria en informática

Nombre de la materia: Programación con objetos I

Responsable de la asignatura y equipo docente: Gonzalez Tulian, Gerardo Martin; Spigariol, Lucas

Cuatrimestre y año: 2^{do} del 1^{er} Año

Carga horaria semanal: 8 horas

Programa N°:

Código de la materia en SIU: 753

PROGRAMACIÓN CON OBJETOS I

1. Fundamentación

La asignatura Programación con objetos I pertenece al área Algoritmos y Lenguajes y se dicta cuatrimestralmente en el primer año según el plan de estudios vigente, además tiene como correlativas precedentes a la asignatura Introducción a la Programación, y como espacio curricular subsiguiente a la asignatura Programación con objetos II.

La asignatura reviste vital importancia para el/la estudiante ya que en los entornos dinámicos y cambiantes donde participan sistemas informáticos los problemas requieren soluciones de sistemas de información que pueden ser de muy variado tipo y con características muy distintas unos de otros. Es indispensable que el/la estudiante comprenda y conozca el enfoque de la programación con objetos y que pueda aplicarlo en la solución de los problemas.

2. Propósitos y/u objetivos

Propósitos

- Introducir formalmente a los/as estudiantes en la programación con objetos, que conozcan y comprendan sus conceptos fundamentales y ventajas.
- Brindar las herramientas necesarias para que los/as estudiantes adquieran destreza modelando y desarrollando prácticas concretas de complejidad simple, moderada y avanzada
- Instalar la noción de cultura colaborativa como fundamento de la comunidad UNAHUR.

Objetivos

Que el/la estudiante:

- Comprenda la representación interna de los objetos, que identifique su estado interno, comportamiento e identidad.
- Pueda modelar y desarrollar situaciones concretas simples, utilizando correctamente los conceptos de mensaje, comportamiento, clases, responsabilidad, delegación e interfaces.
- Entienda cómo se relacionan entre sí los objetos y cómo se representan en el estado interno de un objeto.
- Comprenda y adquiera destreza en el manejo de colecciones.
- Entienda el concepto de polimorfismo y lo utilice correctamente desarrollando situaciones prácticas y concretas.
- Entienda el concepto de herencia simple, identifique las formas correctas e incorrectas de su uso y pueda aplicarlo modelando y desarrollando situaciones prácticas y concretas.
- Desarrolle pruebas unitarias y entregue su código con su correspondiente prueba unitaria, entendiendo que mejoran la calidad del programa y que pueden ser utilizados como documentación.
- Adquiera nociones básicas como tratar los errores de un programa. Entienda qué es una excepción y qué no, y como pueden manejarse.
- Se enfrente a situaciones no intuitivas y adquiera pericia para modelar y desarrollar soluciones donde resulta apropiado relacionar múltiples objetos entre sí.
- Adquiera destreza en el pensamiento en términos de interfaces como contrato al que se comprometen ciertos objetos.

3. Programa sintético:

Conceptos fundantes del paradigma: objeto y mensaje. Concepto de polimorfismo en objetos, comprensión de las ventajas de aprovecharlo. Protocolo/interfaz, concepto de tipo en objetos, comprensión de que un objeto puede asumir distintos tipos. La interfaz como contrato al que se compromete el objeto modelado, posibilidad de reforzar ese contrato. Estado en el paradigma de objetos: referencias, conocimiento, estado interno. Métodos, clases, herencia simple, method lookup. Conceptos de responsabilidad y delegación. Colecciones: conceptualización como objetos, caracterización a partir de los conceptos de protocolo y responsabilidad, protocolo, acceso a sus elementos. Testeo automático y repetible. Nociones básicas sobre manejo de errores. Interrupción del flujo de ejecución: modelado mediante estructuras de control, concepto de excepción.

4. Programa analítico

4.1 Organización del contenido:

UNIDAD I – PROGRAMA, OBJETO Y MENSAJE
<p>Definición de programa como objetos que se comunican entre ellos.</p> <p>Objetos como representaciones computacionales de entes que exhibe un comportamiento.</p> <p>Definición de un objeto. Noción de estado interno para recordar valores, concepto de atributo.</p> <p>Noción de mensaje como punto de acceso al comportamiento exhibido por los objetos.</p> <p>Conocimiento elemental de método como código que se ejecuta cuando un objeto recibe un mensaje. Comprobación del comportamiento y estado interno de los objetos.</p>
UNIDAD II – REFERENCIAS E INTERACION ENTRE OBJETOS, POLIMORFISMO
<p>Relaciones entre objetos que conforman un modelo / programa. Referencias entre objetos como parte del estado interno de los mismos. Necesidad de interacción entre los objetos basadas en el comportamiento. Obtener información de un objeto y cambiar el estado interno.</p> <p>Polimorfismo. Explicación del polimorfismo modelando situaciones donde aparecen distintos objetos con los que un tercero puede interactuar indistintamente que necesite saber si con quien está interactuando.</p> <p>Igualdad e Identidad, diferencia.</p>
UNIDAD III – COLECCIONES Y CLOUSURES
<p>Colección como un objeto que agrupa y mantiene referencias a una cantidad indeterminada de otros objetos. Listas y conjuntos, definición y diferencias entre ambos. Comportamiento esperado.</p> <p>Noción de Clousure como porción de código que se puede convertirse en un objeto y que puede diferir su ejecución.</p> <p>Mensajes básicos que entienden las colecciones. Mensajes sin y con efecto. Comandos, Condiciones, transformaciones y otros mensajes útiles sin y con efecto.</p> <p>Mensajes que solo entienden la listas sin y con efecto.</p> <p>Mensajes que solo entienden los conjuntos sin y con efecto.</p>
UNIDAD IV – CLASES
<p>Motivado por la necesidad de clasificar los objetos, agrupándolos en función a sus semejanzas.</p>

Concepto de clase como patrón o modelo para crear objetos que describe completa y detalladamente la estructura de información y comportamiento que tendrá todo objeto de la clase.

Instanciación como creación de objetos a partir de una clase. Constructores de clase. Pseudovariable self.

UNIDAD V – HERENCIA SIMPLE

Definición de nuevas clases basadas en clases existentes. Superclase y subclase que permite agregar nuevas variables y métodos y modificar los métodos heredados.

Mecanismo de búsqueda con y sin herencia. Redefinición de métodos. Pseudovariable super.

Herencia como herramienta que potencia y amplía el polimorfismo.

Noción de Clase Abstracta. Template Method.

UNIDAD VI – PRUEBA UNITARIA

Importancia de las pruebas unitarias en la calidad del desarrollo de programas. Utilización como documentación del programa.

Elementos que conforman una prueba: objetos de prueba, configuración de escenarios, aserciones.

UNIDAD VII – ERRORES Y EXCEPCIONES

Errores básicos del tipo objetos que no entienden mensajes y no permite la relación entre objetos.

Errores relacionados con las condiciones y la posibilidad de control de estos.

Definición de excepción como evento que altera el flujo normal de mensajes entre los objetos. Noción de propagar la excepción hacia arriba.

4.2 Bibliografía y recursos obligatorios:

Bibliografía obligatoria:

Guías de ejercicios y material adicional generado por el grupo docente disponible en página de la materia: <https://obj1-unahur.github.io/apuntes>

Bibliografía de consulta:

- Barnes, D., Kölling, M. (2014). Programación Orientada a Objetos con Java usando Blue, 5ta edición. Madrid, España: Pearson.

- Fontenla, C. y Páez, N. (2015). *Programación Orientada a Objetos*. Recuperado de <https://legacy.gitbook.com/book/nicopaez/poo/details>.
- Izquierdo, L.R. *Introducción a la Programación Orientada a Objetos*. Recuperado de <http://luis.izqui.org/resources/ProgOrientadaObjetos.pdf>.
- Wirfs- Brock, R., Wilkerson, B. y Wiener, L. (1990). *Designing Object-Oriented Software*. Michigan, USA: Prentice Hall.
- Meyer, B. (1997). *Object Oriented Software Construction, 2nd edition*. California, USA: Prentice Hall.

Herramientas

Wollok: <http://www.wollok.org>

Se trata de un lenguaje de programación acompañado de un entorno integrado de programación, desarrollados en Argentina por docentes de distintas universidades públicas.

Wollok combina un enfoque pedagógico orientado a estudiantes que están haciendo sus primeras armas en programación orientada a objetos, y con poca experiencia en programación. Al mismo tiempo, las características del lenguaje y del entorno de programación son similares a las de lenguajes y herramientas ampliamente utilizados en la industria, como ser los lenguajes Java y JavaScript, y el entorno Eclipse.

5. Metodología de enseñanza:

La asignatura está constituida por 7 unidades temáticas, las cuales se dictarán durante un cuatrimestre en clases teóricas y prácticas ,50% en formato presencial y 50% en formato virtual de clase sincrónica, con resolución de problemas a cargo de los/las alumnos/as supervisados por los/las docentes. Las clases están organizadas para integrar los contenidos teóricos en la solución de problemas prácticos presentados a los/las estudiantes. El nivel de complejidad de los problemas aumenta en función a la presentación de los contenidos teóricos de cada unidad.

El objetivo de esta metodología busca que el saber de los conceptos introducidos en cada clase se refleje en el saber y hacer práctico de la solución de un problema planteado. Se presentan los conceptos teóricos, se complementa con videos teórico/práctico y lecturas adicionales recomendadas, y se verifica en el hacer práctico que se apliquen los conceptos presentados.

Cada problema es presentado y discutido teóricamente, y luego es resuelto por los/las estudiantes directamente mediante la implementación de programas. Esta estrategia unifica los conceptos, el modelado y solución del problema, y la implementación como clase de laboratorio.

La carga horaria de la materia permite la discusión teórica incentivando el espíritu de colaboración e investigación, como así también la implementación de las soluciones modeladas en las computadoras, donde cada estudiante trabaja con la asistencia de los/las docentes y de sus compañeros/as.

Por último, la organización propuesta para la cursada está fuertemente basada en la existencia de una serie de guías de ejercicios, que están graduados de forma de acompañar la evolución de los/las estudiantes y la secuencia pedagógica propuesta.

6. Plan de trabajo en el campus:

El campus será utilizado como el canal formal de comunicación con los/as estudiantes y será el punto de partida para llegar a otros sitios donde se alojará todo lo necesario para la cursada.

La materia cuenta con un sitio propio <https://obj1-unahur.github.io/> al que se podrá acceder desde el campus universitario y será utilizado en forma complementaria al campus universitario.

El programa y cronograma será publicado en el campus universitario en la sección solapa "General". Las notas serán publicadas en la solapa "Notas"

7. Evaluación y régimen de aprobación

7.1 Aprobación de la cursada

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos. Todas las instancias evaluativas deberán tener una instancia de recuperatorio. Podrán acceder a la administración de esta modalidad solo aquellos y aquellas estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial.

Siempre que se realice una evaluación de carácter recuperatorio, la calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

El/La alumno/a deberá poseer una asistencia no inferior al 75% en las clases presenciales.

En cuanto a la cursada de la materia en modalidad virtual se requerirá que el/la estudiante ingrese al aula virtual como mínimo una vez por semana.

7.2 Aprobación de la materia

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

Promoción directa: tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.

Evaluación integradora: tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos estudiantes que hayan aprobado la cursada con una nota de entre cuatro (4) y seis (6) puntos.

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrado, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia.

La nota obtenida se promediará con la nota de la cursada.

Examen final: Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.

7.3 Criterios de calificación

La calificación de cada evaluación se determinará en la escala 0 a 10, con los siguientes valores: 0, 1, 2 y 3: insuficientes; 4 y 5 regular; 6 y 7 bueno; 8 y 9 distinguido; 10 sobresaliente.

8. Cronograma

CRONOGRAMA		
SEMANA	UNIDAD	TEMA TEÓRICO Y EJERCITACIÓN PRÁCTICA
01 (Presencial)	I	Introducción a la Materia. Conceptos de programa objeto y mensaje. Definición de un objeto, implementación de métodos y envío de mensajes. Ejercitación con de diagnóstico y primer ejercicio práctico de introducción.

02 (Virtual)	II	Interrelación entre los objetos. Introducción al polimorfismo. Ejercitación de 2 objetos con los mismos métodos y un tercer objeto que interactúa con los dos primeros en forma indistinta.
03 (Virtual)	III	Introducción a las colecciones y clousures. Listas y Conjuntos, operaciones básicas agregar, remover y modificar objetos de las colecciones. Ejercitación de ejemplos triviales por consola. Ejercicio donde un objeto mantiene una colección, recorrer la colección y enviar mensajes polimórficos a cada objeto.
04 (Presencial)	III	Comandos, Condiciones, transformaciones y otros mensajes útiles sin y con efecto. Mensajes que solo entienden las listas sin y con efecto. Mensajes que solo entienden los conjuntos sin y con efecto. Ejercicio que implemente el recorrido, comandos, filtros y transformaciones sobre las colecciones.
05 (Virtual)	III	Más ejercitación práctica sobre el manejo de colecciones. Presentación de un trabajo práctico que integre los conceptos hasta la unidad III.
06 (Virtual)	IV y VI	Introducción al concepto de Clase. Referencias entre objetos. Creación de instancias, uso de correcto del self. Introducción a las pruebas unitarias Ejercicio práctico donde se presente la necesidad de creación de varios objetos de un mismo tipo y complementa el uso de la colección.
07 (Presencial)	IV y VI	Creación dinámica de instancias, repaso de referencias usando instancias. Repaso de prueba unitarias. Polimorfismo entre instancias de distintas clases. Ejercicio práctico consolidador de los temas vistos hasta el momento
08 (Presencial)	Parcial I	Consolidación de los temas vistos hasta el momento. Ejercicios Prácticos. Primer parcial
09 (Virtual)	V	Introducción a la herencia simple. Redefinición de métodos. Ejercitación básica que contempla varias clases, superclase y subclase.
10 (Virtual)	V	Más ejercitación con Redefinición de métodos. Pseudovariable super. Herencia como herramienta que potencia y amplía el polimorfismo. Desarrollo del trabajo práctico integrador
11 (Presencial)	V	Consolidación del concepto de Herencia Simple. Casos complejos de Herencia. Mecanismos de búsqueda con herencia. Práctica y desarrollo de ejercicios complejos que contemplen herencia y colecciones.

12 (Virtual)	VI y VII	Pruebas unitarias con varios escenarios. Manejo de Errores y Excepciones. Practica con un ejercicio integrador
13 (Virtual)	Todas	Practica con ejercicios complejos. Desarrollo de prueba unitaria con varios escenarios. Asignación de responsabilidades.
14 (Presencial)	Todas	Consolidación de los temas vistos hasta el momento. Ejercicios Prácticos. Segundo parcial
15 (Presencial)	Todas	Cierre de la materia. Recuperatorio del primer parcial
16 (Presencial)	Todas	Cierre de la materia. Recuperatorio del segundo parcial