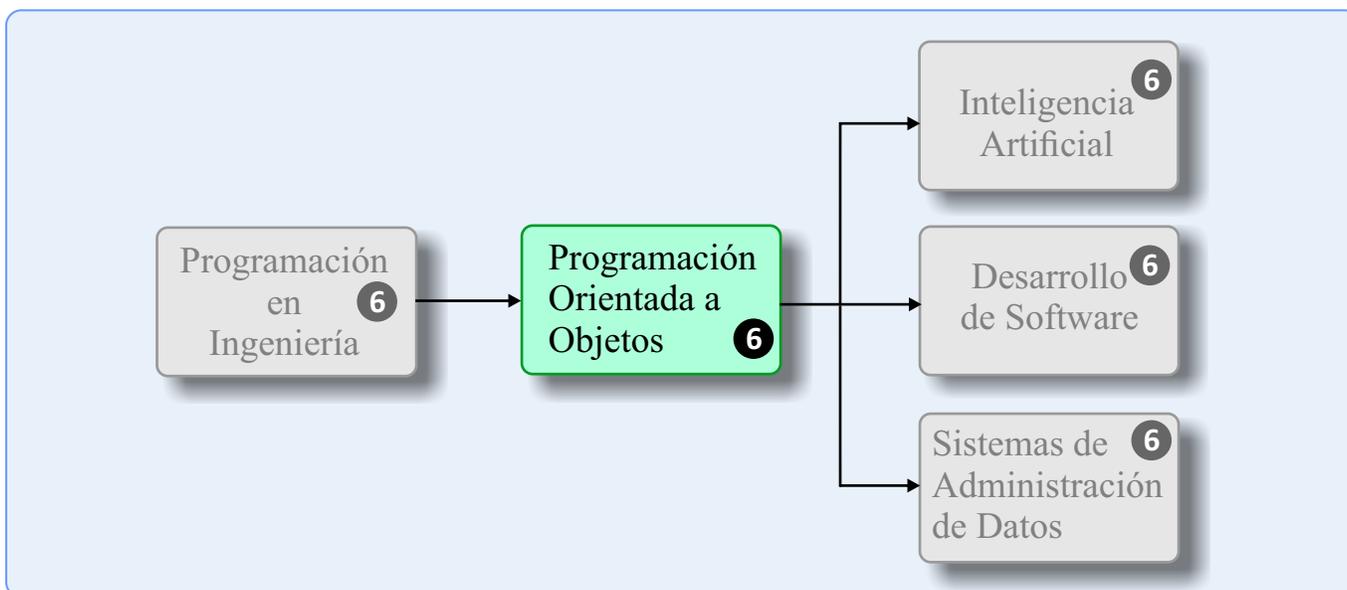




IILI06107

Programación Orientada a Objetos

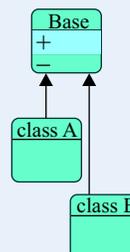
Lunes y Jueves: 2:00 p.m. - 4:00 p.m., 310.



1. Datos descriptivos

1. 1. Descripción del curso

El alumno aprenderá las bases teóricas de la Programación Orientada a Objetos para el diseño de clases a fin de resolver problemas en ingeniería usando el comando **class**. El alumno aprenderá como usar en forma apropiada los comandos de acceso en una clase: **public**, **private** y **protected**. Así mismo se explora la forma apropiada de usar la herencia, las clases abstractas y las interfaces. El curso intenta explicar las buenas prácticas en la planeación, organización e implementación de la Programación Orientada a Objetos. El objetivo del curso no es el de enseñar al estudiante a utilizar alguna tecnología o software específico, en su lugar el curso presenta la teoría general que puede ser aplicada a las distintas tecnologías en hardware y software. Para la impartición de este curso se puede usar cualquier lenguaje que soporte la programación orientada objetos tal como: C++, Java o C#.



Importante: El uso de teléfonos celulares está estrictamente prohibido en clase, así estos deben fijarse en el modo de vibrador y **colocarse dentro de la mochila**.

1.2. Prerrequisitos

El alumno debe haber cursado la UDA ILLI06104 **Programación en Ingeniería**. Este curso requiere que el alumno tenga una computadora portátil. Antes de iniciar el curso, el alumno debe ser capaz de: leer en el idioma inglés, usar con seguridad las sentencias básicas de programación (if, if-else, for, while, etc.) y las funciones. Además el estudiante debe ser capaz de poder instalar y configurar programas en su computadora. Si usted no cumple con los requisitos para tomar este curso, se le recomienda que se "de baja" del curso y solo se inscriba hasta que tenga los requisitos.



Para este curso el alumno puede usar algún ambiente de desarrollo tal como: Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc. Estudiantes de la Universitarios pueden descargar Microsoft Visual Studio de www.dreamspark.com



Este curso prepara al estudiante para el ambiente de trabajo. En este sentido el alumno tendrá la oportunidad de trabajar bajo la presión de fechas de entrega y el estudiante pasará realizando actividades en la computadora por varias horas en casa. Así mismo el estudiante practicará como comunicarse en forma efectiva y positiva con los demás. Si a usted le molesta trabajar en la computadora por favor darse de baja.

1.3. Facilitador

Dr. Sergio Eduardo Ledesma Orozco, profesor investigador de la Carrera de Ingeniería en Sistemas Computacionales perteneciente al Departamento de Ingeniería Electrónica.



1.4. Horas de Asesoría

Los asistentes al curso podrán contar con atención continua en la dirección electrónica: selo@ugto.mx



2. Sentido de la materia en el Programa Educativo de Sistemas Computacionales.

Este curso presenta las bases para que el estudiante diseñe sus programas usando la programación orientada objetos en lugar de usar la programación estructurada. El estudiante debe poder reemplazar el uso de las estructuras y las funciones y en su lugar diseñar clases.

3. Competencias que los participantes desarrollarán en la materia:

Diseña e implementa un programa de computadora usando la programación orientada a objetos para resolver problemas en ingeniería, investigación o negocios. En esta unidad el estudiante utilizará los lenguajes de C++, Java, C# u otro lenguaje que soporte la programación orientada a objetos.

La Unidad de Aprendizaje incide de manera directa en la formación de la competencia genérica institucional: Maneja ética y responsablemente las tecnologías de la información para agilizar sus procesos académicos y profesionales de intercomunicación.

Contribuye a la competencia específica del programa:

CE1. Planifica, desarrolla, implementa, da mantenimiento y administra sistemas de información para realizar de manera eficiente el procesamiento de información con el uso de técnicas modernas de tratamiento.

CE4. Diseña, instala, programa y proporciona soporte técnico a sistemas de información basados en dispositivos de hardware programable para generar valor en el procesamiento de información de las organizaciones.

CE5. Diseña, implementa y da mantenimiento a sistemas inteligentes incluyendo sus aplicaciones a los sistemas de información, la informática industrial y la robótica móvil.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

CE8. Participa en proyectos de investigación que lo orientan hacia programas de posgrado de calidad en el área de los sistemas computacionales.

4. Contenidos

4.1. Estructura

El curso se divide en tres fases:

1. Repaso de: las sentencias de control de un programa (if, if-else, for, while, etc) y elementos de Interface Gráfica de usuario (caja de texto, botón, radiobutton checkbox, etc.)
2. Teoría de la programación orientada a objetos (class, public, private, protected, friend, etc.)
3. Aplicación de la programación orientada a objetos a problemas en ingeniería.

4.2. Epígrafes de la materia:

1. Repaso de las sentencias de control y de los elementos de interface gráfica de usuario
2. Diseño e implementación de clases: variables miembro y funciones miembro.
3. Acceso a miembros de una clase.
4. Manejo de excepciones (bloques try-catch).
5. Propiedades de una clase (set y get).
6. Constructor de una clase y constructor de copia.
7. Herencia.
8. Operadores (operator=, operator+, casting operators, subscript operator, etc.)
9. Clases amigo (friend).
10. Funciones virtuales y clases abstractas.
11. Variables locales estáticas y variables miembro estáticas.
12. Funciones miembro estáticas.
13. Diseño e implementación de Interfaces.
14. Espacios de nombres (Namespaces).
15. Aplicación de la programación orientada a objetos en ingeniería

Bibliografía básica:

1. B. Stroustrup, (2013) The C++ Programming Language, 4th Edition.
2. A. Koenig, B. E. MOO, (2000) Accelerated C++: Practical Programming by Example.
3. N. M. Josuttis, (2012) The C++ Standard Library: A Tutorial and Reference.
4. B. Stroustrup, (2013) A Tour of C++ (C++ In-Depth Series).
5. R. Lafore, (2001) Object-Oriented Programming in C++ (4th Edition)
6. A. V. Dam, (2005) Object-Oriented Programming in Java: A Graphical Approach.
8. J. Bloch, (2008) Effective Java (2nd Edition).

Todas las notas de clase se encuentran disponibles en forma electrónico dentro del tutorial de Wintempla. Para instalar la notas de clase en su computadora usted necesita Microsoft Visual Studio. Una vez que Microsoft Visual Studio esta instalando use el siguiente enlace para instalar las notas de clase:

<https://visualstudiogallery.msdn.microsoft.com/4dfcaf79-9f69-4fea-87b6-ec222f1d0fdd>

5. Metodología y distribución de la carga de trabajo

El curso esta organizado para promover la constante participación del estudiante en discusiones y trabajo en equipo. El profesor presentará la información básica de cada tema, entonces los estudiantes trabajarán en forma individual en problemas relacionados con la materia. Por cada hora de clase, el alumno debe realizar una hora y cinco minutos trabajo autónomo, esto implica que todos los días de clase se asigna tarea. La tarea puede consistir en investigar de un tema específico o resolver un problema usando la programación orientada a objetos. Por favor apague su teléfono celular antes de entrar a clase.

Actividades	Horas presenciales	Trabajo autónomo	Total
Clases teóricas	18	20	38
Exposiciones	4	8	12
Ejercicios prácticos	50	50	100
Total	72	78	150

6. Evaluación

6.1. Tareas (20%)

La tarea se puede entregar solamente el día asignado para su entrega exactamente al inicio de clase. **En caso de que el alumno falte el día de la entrega de la tarea, la tarea no se tomará en cuenta aún cuando la tarea se haya enviado por correo electrónico.** La única excepción para entregar la tarea en la clase siguiente es la muerte de un hermano, hermana, padre o madre, tío, hermanos o padres del papá o la mamá y cuando la falta se justifique en la clase siguiente.

El profesor asignará tareas que el estudiante puede resolver por su cuenta. Es responsabilidad del estudiante asistir a las clases y poner atención a lo que el profesor explica en clase. En caso de faltar a clase, el estudiante debe preguntar a otros estudiantes sobre la tarea asignada. **Es normal que si un estudiante falta a clases, también tendrá problemas para hacer las tareas, de igual forma también tendrá problemas para entender el material de clases en el futuro, en este caso, el estudiante es responsable de ponerse al corriente.**

Los archivos fuente (*.cpp, *.java, *.cs), sin ningún tipo de compresión, deben de enviarse por correo electrónico antes del inicio de clase a la dirección selo@ugto.mx. El asunto del mensaje es 107POO y dentro del cuerpo del mensaje debe escribirse su nombre, su NUA y una breve descripción o reflexión del proyecto asignado.

6.2. Exámenes (55%)

Primer examen parcial el 27 de Septiembre a la hora de clase **27.5%**. Examen final el 3 de Diciembre a la hora de clase **27.5%**. El examen sólo se puede presentar en la fecha y hora asignada. No hay ninguna excepción, ni siquiera si el estudiante discute con el profesor historias tristes y conmovedoras.

6.3. Asistencia (10%)

El punto de asistencia se otorga cuando el alumno falta cero, una o dos veces y se llega dentro de los primeros 10 minutos del inicio de la clase. **Una falta se puede justificar por escrito solamente la clase siguiente a la falta.** Se pueden justificar faltas por razones médicas o por muerte de un hermano, hermana, padre o madre, tío, hermanos o padres del papá o la mamá.

A fin de tratar de mantener un sentido de igualdad con otros estudiantes, no es posible justificar una falta discutiendo problemas personales y familiares con el profesor. Tampoco es posible justificar una falta el día de entrega de la calificación final. **De hecho el único momento para justificar una falta es la clase siguiente a la falta.**

6.4. Proyecto Final (15%)

El proyecto se presenta a la hora de clase los días 26 y 29 de Noviembre junto con un reporte de dos hojas impreso en papel, no existen excepciones, ni siquiera si el estudiante discute con el profesor historias tristes y conmovedoras. De igual forma el día de entrega del proyecto o el día de entrega de calificaciones finales **queda prohibido entregar tareas retrasadas o justificar inasistencias. Una vez terminada la hora de entrega de proyectos, el estudiante no puede pasar a la oficina del profesor a discutir con él ningún tipo de historias (especialmente historias tristes y conmovedoras).**

El proyecto debe presentarse aún cuando la computadora se descompuso la noche anterior o aún cuando el disco duro de la computadora dejó de funcionar inesperadamente.

6.5. Portafolio del Estudiante (Es un requisito para entregar el proyecto final)

Se entrega un CD con tres carpetas:

- Tareas: se encuentra una carpeta por cada tarea; en cada carpeta se incluyen los archivos de esa tarea.
- Ejercicios : se incluye una carpeta por cada ejercicio hecho en clase.
- Proyecto: se incluyen los archivos del proyecto final.

El CD se entrega en el momento de presentar el proyecto final, no existe ninguna excepción para entregar el CD otro día. En el directorio raíz del CD se encuentra un archivo de texto llamando leeme.txt con el nombre y NUA del estudiante. Además, el CD tiene una etiqueta con:

1. Nombre del alumno,
2. El NUA del alumno,
3. Nombre del curso,
4. Periodo (ejemplo: agosto-diciembre 2015),
5. Nombre del profesor.



Tareas, proyectos y exámenes (archivos en el portafolio del estudiante)

El sistema de competencias requiere del trabajo autónomo del estudiante. En caso de encontrar evidencia de que un alumno entregue tareas o proyectos que no fueron realizados por el propio estudiante, el profesor turnará el caso a la **Comisión de Honor y Justicia** de la División de Ingenierías. Se tiene precedencia que esta comisión ha dictaminado en el pasado la suspensión del estudiante por un periodo escolar o por un año. En los documentos electrónicos es muy fácil demostrar cuando el trabajo autónomo fue realizado por otro estudiante. Se le pide al estudiante esforzarse por realizar sus actividades de trabajo autónomo y en caso de no tener las habilidades para realizar estas actividades en forma correcta, el estudiante debe proporcionar la adecuada retroalimentación al profesor.

Recuerde que el profesor y el estudiante forman un equipo en la que la única meta es el aprendizaje. El estudiante debe mantener una actitud positiva y en lugar de quejarse y encontrar culpables, él debe buscar la forma de solucionar el problema. El estudiante debe evitar compararse con otros estudiantes de la misma institución u otra institución educativa; en su lugar él debe concentrarse en disfrutar de las actividades escolares y en alcanzar sus metas.

7. Bibliografía para ampliar el tema:

1. <http://www.cplusplus.com>
2. <http://msdn.microsoft.com>
3. <http://docs.oracle.com/javase/tutorial/>

Tema 1

(Semana 1 y 2)

Repaso de las sentencias de control y de los elementos de interface gráfica de usuario

1. Sentido del tema

A fin de establecer un fundamento robusto en el diseño e implementación de programas para computadora, el estudiante repasará los conceptos básicos de la programación incluyendo aquellos de la programación estructurada para prepararse para la programación orientada a objetos.

2. Epigrafs del tema

1. Declaración de variables: enteras, de punto flotante, booleanas, tiempo, y color.
2. Uso de las sentencias de control: if, if-else, for, break, continue, return, while, do-while y switch.
3. Uso de las funciones en un programa.
4. Elementos de Interface Gráfica de Usuario (GUI): caja de texto, botón, radio botón, checkbox, lista desplegable, grupo de pestañas, barra de herramientas, vista de lista, etc.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE1. Planifica, desarrolla, implementa, da mantenimiento y administra sistemas de información para realizar de manera eficiente el procesamiento de información con el uso de técnicas modernas de tratamiento.

CE4. Diseña, instala, programa y proporciona soporte técnico a sistemas de información basados en dispositivos de hardware programable para generar valor en el procesamiento de información de las organizaciones.

CE5. Diseña, implementa y da mantenimiento a sistemas inteligentes incluyendo sus aplicaciones a los sistemas de información, la informática industrial y la robótica móvil.

6. Dificultades principales

El estudiante no ha practicado recientemente la programación y no recuerda muy bien como hacer programas.

Tema 2

(Semana 3)

Diseño e implementación de clases: variables miembro y funciones miembro

1. Sentido del tema

A fin de simplificar el diseño e implementación de programas, el estudiante comenzará a usar el comando **class** para resolver problemas sencillos en ingeniería. El estudiante puede usar los lenguajes: C++, C# o Java. De igual forma el estudiante entenderá la diferencia entre una clase y un objeto.

2. Epigrafs del tema

1. Conceptos básicos de una clase.
2. Sintaxis de una clase en C++, Java o C#.
3. Variables miembro de una clase y su inicialización.
4. Funciones miembro de una clase.
5. Crear y usar objetos de una clase.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE1. Planifica, desarrolla, implementa, da mantenimiento y administra sistemas de información para realizar de manera eficiente el procesamiento de información con el uso de técnicas modernas de tratamiento.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

CE8. Participa en proyectos de investigación que lo orientan hacia programas de posgrado de calidad en el área de los sistemas computacionales.

6. Dificultades principales

El estudiante está acostumbrado a usar la programación estructurada.

Tema 3

(Semana 4)

Acceso a miembros de una clase (public, private y protected)

1. Sentido del tema

El estudiante entenderá como limitar el acceso a las variables y funciones de una clase para proteger la integridad de los objetos. Este concepto es muy importante porque lleva la programación a una dimensión más real de como los objetos interactúan en la vida cotidiana.

2. Epigrafs del tema

1. Comando **public**.
2. Comando **private**.
3. Comando **protected**.
4. Ejemplos y problemas.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE1. Planifica, desarrolla, implementa, da mantenimiento y administra sistemas de información para realizar de manera eficiente el procesamiento de información con el uso de técnicas modernas de tratamiento.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

CE8. Participa en proyectos de investigación que lo orientan hacia programas de posgrado de calidad en el área de los sistemas computacionales.

6. Dificultades principales

El estudiante no tiene una idea clara cual es la ventaja de la programación orientada a objetos.

Tema 4

(Semana 5)

Manejo de excepciones (bloques try-catch)

1. Sentido del tema

El estudiante entenderá y usará el bloque **try-catch** para mejorar la calidad y estabilidad de un programa de computadora. El estudiante debe identificar la necesidad de las excepciones y de como atrapar estas excepciones para identificar problemas en un programa durante su ejecución.

2. Epigrafs del tema

1. El bloque **try-catch**.
2. Ejemplos y problemas.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE1. Planifica, desarrolla, implementa, da mantenimiento y administra sistemas de información para realizar de manera eficiente el procesamiento de información con el uso de técnicas modernas de tratamiento.

6. Dificultades principales

Usualmente el estudiante no usado este tipo de bloques en programas previos.

Tema 5

(Semana 5)

Propiedades de una clase (set y get)

1. Sentido del tema

La programación orientada a objetos trata de representar en forma de código los objetos usados en la vida cotidiana, de este modo el estudiante podrá identificar las propiedades de los objetos de una clase y será capaz de implementar las propiedades de este.

2. Epigrafs del tema

1. Variables miembro privadas.
2. Funciones del tipo "get".
3. Funciones del tipo "set" (o "put").
4. Propiedades.
5. Ejemplos y problemas.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE1. Planifica, desarrolla, implementa, da mantenimiento y administra sistemas de información para realizar de manera eficiente el procesamiento de información con el uso de técnicas modernas de tratamiento.

CE5. Diseña, implementa y da mantenimiento a sistemas inteligentes incluyendo sus aplicaciones a los sistemas de información, la informática industrial y la robótica móvil.

6. Dificultades principales

El estudiante tiene problemas para recordar la sintaxis correcta para declarar una propiedad.

Tema 6

(Semana 6)

Constructor de una clase y constructor de copia

1. Sentido del tema

El estudiante podrá identificar cuando un objeto se construye y cuando un objeto se clona (se crea igual a otro objeto existente). De igual forma el estudiante deberá ser capaz de diseñar e implementar los constructores una clase incluyendo el constructor de copia. Finalmente, el estudiante debe entender cuando y como se destruyen los objetos.

2. Epigrafs del tema

1. Constructor de defecto.
2. Constructores múltiples.
3. Constructor de copia.
4. Ejemplos y problemas.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE1. Planifica, desarrolla, implementa, da mantenimiento y administra sistemas de información para realizar de manera eficiente el procesamiento de información con el uso de técnicas modernas de tratamiento.

CE5. Diseña, implementa y da mantenimiento a sistemas inteligentes incluyendo sus aplicaciones a los sistemas de información, la informática industrial y la robótica móvil.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

6. Dificultades principales

El estudiante tiene problemas para identificar cuando se manda llamar el constructor.

Tema 7

(Semana 7)
Herencia

1. Sentido del tema

El estudiante podrá re-usar clases previamente creadas usando la herencia para crear clases nuevas. Primero el estudiante debe diseñar un conjunto de clases, segundo él debe ser capaz de implementar su diseño usando un lenguaje de programación.

2. Epigrafs del tema

1. Herencia.
2. Ejemplos y problemas.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE1. Planifica, desarrolla, implementa, da mantenimiento y administra sistemas de información para realizar de manera eficiente el procesamiento de información con el uso de técnicas modernas de tratamiento.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

CE8. Participa en proyectos de investigación que lo orientan hacia programas de posgrado de calidad en el área de los sistemas computacionales.

6. Dificultades principales

El estudiante tiene problemas para llevar a la práctica la herencia en la creación de sus programas.

Tema 8

(Semana 8)

Operadores (operator=, operator+, casting operators, subscript operator, etc.)

1. Sentido del tema

El estudiante podrá identificar las operaciones que se pueden hacer entre los objetos de una clase a fin de diseñar e implementar clases con operadores. Estos operadores son usados para hacer un uso más natural de los objetos de una clase.

2. Epigrafs del tema

1. operator+ y operator-.
2. operator*, operator/ y operator%.
3. operator=.
4. Subscript operator para arreglos.
5. Casting operators para convertir objetos.
6. Ejemplos y problemas.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE1. Planifica, desarrolla, implementa, da mantenimiento y administra sistemas de información para realizar de manera eficiente el procesamiento de información con el uso de técnicas modernas de tratamiento.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

CE8. Participa en proyectos de investigación que lo orientan hacia programas de posgrado de calidad en el área de los sistemas computacionales.

6. Dificultades principales

El estudiante entiende para que se usan pero tiene problema para implementar en forma correcta estos operadores.

Tema 9

(Semana 9)

Clases amigo (friend)

1. Sentido del tema

En algunos casos los comandos **public**, **private** y **protected** no permiten expresar en forma precisa los permisos entre las variables y funciones de una clase. El estudiante deberá poder identificar cuando se debe usar el comando **friend** para cambiar el acceso a los miembros de una clase.

2. Epigrafs del tema

1. Clases amigo.
2. Ejemplos y problemas.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE1. Planifica, desarrolla, implementa, da mantenimiento y administra sistemas de información para realizar de manera eficiente el procesamiento de información con el uso de técnicas modernas de tratamiento.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

CE8. Participa en proyectos de investigación que lo orientan hacia programas de posgrado de calidad en el área de los sistemas computacionales.

6. Dificultades principales

El estudiante tiene problemas para usar clases amigo en sus programas.

Tema 10

(Semana 9)

Funciones virtuales y clases abstractas

1. Sentido del tema

En algunos casos una idea no se encuentra completamente definida, este tipo de ideas se conocen como abstractas porque aún faltan algunos detalles para que puedan llevarse al mundo real. De igual forma las clases abstractas son clases que no se encuentran completamente definidas.

2. Epigrafs del tema

1. Funciones virtuales.
2. Ideas abstractas.
3. Clases abstractas.
4. Convirtiendo una clase abstracta en una clase regular.
5. Ejemplos y problemas.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE5. Diseña, implementa y da mantenimiento a sistemas inteligentes incluyendo sus aplicaciones a los sistemas de información, la informática industrial y la robótica móvil.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

CE8. Participa en proyectos de investigación que lo orientan hacia programas de posgrado de calidad en el área de los sistemas computacionales.

6. Dificultades principales

El estudiante no entiende el verdadero poder de una idea abstracta, de igual forma no le queda claro cual es el propósito de una clase abstracta.

Tema 11

(Semana 10)

Variables locales estáticas y variables miembro estáticas

1. Sentido del tema

En algunos casos existe la necesidad de tener variables que retienen su valor entre llamadas a una función. En otros casos, se tiene la necesidad de compartir una variable entre todos los objetos de una clase. Las variables estáticas cambian el ciclo de vida de una variable más allá de las variables tradicionales.

2. Epígrafes del tema

1. Definiendo y usando variables locales estáticas.
2. Definiendo y usando variables miembro estáticas.
3. Ejemplos y problemas.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE5. Diseña, implementa y da mantenimiento a sistemas inteligentes incluyendo sus aplicaciones a los sistemas de información, la informática industrial y la robótica móvil.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

CE8. Participa en proyectos de investigación que lo orientan hacia programas de posgrado de calidad en el área de los sistemas computacionales.

6. Dificultades principales

El estudiante no distingue en forma clara una variable miembro de una variable miembro estática. De igual forma no le queda claro cual es el propósito de una variable local estática.

Tema 12

(Semana 10)

Funciones miembro estáticas

1. Sentido del tema

En algunos casos existe la necesidad de funciones que no requieren un objeto para ejecutarse, estas funciones se conocen como funciones estáticas. El estudiante debe ser capaz de identificar la necesidad de este tipo de funciones, de igual forma el estudiante debe ser capaz de implementar clases con funciones estáticas.

2. Epigrafs del tema

1. Clases con funciones miembro estáticas.
2. Llamando una función miembro estática.
3. Ejemplos y problemáticas.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE5. Diseña, implementa y da mantenimiento a sistemas inteligentes incluyendo sus aplicaciones a los sistemas de información, la informática industrial y la robótica móvil.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

CE8. Participa en proyectos de investigación que lo orientan hacia programas de posgrado de calidad en el área de los sistemas computacionales.

6. Dificultades principales

El estudiante no recuerda como mandar llamar las funciones miembro estáticas.

Tema 13

(Semana 11)

Diseño e implementación de Interfaces

1. Sentido del tema

Uno de los conceptos más interesantes de la programación orientada a objetos es el de las interfaces. Una interface es un conjunto de funciones virtuales que permiten interactuar entre objetos. El estudiante debe identificar las ventajas de usar las interfaces a fin de poder diseñar nuevas interfaces. De igual forma el estudiante debe poder implementar una interface.

2. Epígrafes del tema

1. Concepto y ventajas de una interface.
2. Declarar una interface.
3. Implementar una interface.
4. Ejemplos y problemas.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE5. Diseña, implementa y da mantenimiento a sistemas inteligentes incluyendo sus aplicaciones a los sistemas de información, la informática industrial y la robótica móvil.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

CE8. Participa en proyectos de investigación que lo orientan hacia programas de posgrado de calidad en el área de los sistemas computacionales.

6. Dificultades principales

El estudiante tiene problemas para entender las ventajas de las interfaces en aplicaciones reales.

Tema 14

(Semana 11)

Espacios de Nombres (Namespaces)

1. Sentido del tema

Una vez que se tienen muchas clases es muy difícil recordar todas ellas. A fin de organizar las clases un **namespace** es un espacio que permiten organizar o agrupar clases. El estudiante deberá entender el concepto de **namespace** a fin de que él pueda usarlos en sus programas.

2. Epigrafs del tema

1. Concepto y ventajas de los espacios de nombres.
2. Declarar una interface.
3. Implementar una interface.
4. Ejemplos y problemas.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE5. Diseña, implementa y da mantenimiento a sistemas inteligentes incluyendo sus aplicaciones a los sistemas de información, la informática industrial y la robótica móvil.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

CE8. Participa en proyectos de investigación que lo orientan hacia programas de posgrado de calidad en el área de los sistemas computacionales.

6. Dificultades principales

El estudiante tiene problemas para distinguir entre un **namespace** y una clase.

Tema 15

(Semana 12 a 18)

Aplicación de la programación orientada a objetos en ingeniería

1. Sentido del tema

A fin de poder entender el poder de la programación orientada a objetos, el estudiante realizará múltiples programas de computadora en el área de ingeniería usando la programación orientada a objetos.

2. Epigrafs del tema

1. Conversión de programación estructura a programación orientada a objetos.
2. Resolución de problemas en ingeniería usando la programación orientada a objetos.
3. El modelo de objetos comunes de Microsoft (COM).
4. Las interfaces IUnknown e IDispatch.
5. El explorador de objetos.
6. El comando **#import** para reusar Microsoft Word, Excel, Power Point, Corel Draw, etc.
7. La librería de plantillas activas (ATL) y sus usos.
8. Creando un servidor COM.
9. Usando el comando class para crear simplificar el uso de COM.

3. Materiales para estudiarlo

Computadora portátil, resultados de búsquedas en la Internet, notas de clase, videos, ambiente de desarrollo (IDE) tal como Microsoft Visual Studio, Eclipse, JBuilder, IntelliJ, etc.

4. Método de trabajo aconsejado

Revisión de códigos de programa principalmente clases previamente implementadas. Diseño e implementación de clases para resolver un problema en ingeniería. Uso del ambiente de desarrollo (IDE) para detectar y corregir errores. Creación e inspección de diagramas de clases en UML.

5. Competencias trabajadas

CE1. Planifica, desarrolla, implementa, da mantenimiento y administra sistemas de información para realizar de manera eficiente el procesamiento de información con el uso de técnicas modernas de tratamiento.

CE5. Diseña, implementa y da mantenimiento a sistemas inteligentes incluyendo sus aplicaciones a los sistemas de información, la informática industrial y la robótica móvil.

CE6. Planifica, administra y evalúa proyectos basados en sistemas computacionales desempeñando roles como asesor, consultor y cargos gerenciales.

CE8. Participa en proyectos de investigación que lo orientan hacia programas de posgrado de calidad en el área de los sistemas computacionales.

6. Dificultades principales

El estudiante tiene problemas para dejar de usar la programación estructurada. El estudiante se resiste a cambiar su forma de diseñar e implementar programas.