



**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERIA**

SYLLABUS

Ingeniería Catastral y Geodesia

NOMBRE DEL DOCENTE:

ESPACIO ACADÉMICO (Asignatura):

Programación Orientada por Objetos

Obligatorio (X) : Básico (X) Complementario ()
Electivo () : Intrínsecas () Extrínsecas ()

CÓDIGO: 10

NÚMERO DE ESTUDIANTES:

GRUPO:

NÚMERO DE CREDITOS: 3

TIPO DE CURSO: **TEÓRICO** ☐ **PRACTICO** ☐ **TEO-PRAC:** ☒

Alternativas metodológicas:

Clase Magistral (x), Seminario (), Seminario – Taller (), Taller (x), Prácticas (x), Proyectos tutoriados (), Otro: _____

HORARIO:

DÍA	HORAS	SALÓN

I. JUSTIFICACIÓN DEL ESPACIO ACADÉMICO

Competencias del perfil a las que contribuye la asignatura:	<p>Esta asignatura contribuye al desarrollo de la competencia “<i>Desarrolla Habilidades en el uso de las tecnologías de la información y de la comunicación para la gestión de datos espaciales</i>” y “<i>Demuestra capacidades para asimilar, aplicar, investigar y desarrollar nuevas tecnologías para la gestión de la información espacial</i>” que se encuentra en el dominio de “programación” del área “básicas de ingeniería” del proyecto curricular de Ingeniería Catastral y Geodesia.</p> <p>Identificar, modelar y representar objetos bajo lenguajes de modelado y programación.</p>
Contribución a la formación:	Este espacio académico orienta al estudiante en el uso del paradigma de programación orientado a objetos, brindándole las herramientas para aplicar los principios y características de este paradigma.
Puntos de apoyo para otras asignaturas:	Estructura lógica conceptual basada en el paradigma de programación orientada a objetos.

	<ul style="list-style-type: none"> Herramienta fundamental para Bases de Datos Herramienta fundamental para Programación de Interfaces SIG.
Requisitos previos:	Programación Básica
II. PROGRAMACIÓN DEL CONTENIDO	
OBJETIVO GENERAL	
<p>Presentar al estudiante el paradigma de la programación orientada por objetos, enfatizando en los elementos conceptuales propios del mismo, con el fin de que plantee y aplique modelos de una manera eficiente y natural utilizando un lenguaje de programación orientado a objetos.</p>	
OBJETIVOS ESPECÍFICOS	
<ol style="list-style-type: none"> Determinar los tipos de aplicación y las situaciones en las que se debe aplicar el paradigma orientado a objetos. Comprender, interpretar y analizar el cambio de enfoque en el modo de resolver problemas que supone el uso del paradigma orientado a objetos respecto a otros paradigmas. Aplicar los conceptos del paradigma de programación orientada a objetos tales como: polimorfismo, encapsulamiento, herencia, sobrecarga, funciones virtuales, etc. Dominar estrategias básicas de reutilización como son el uso de librerías y de patrones software. Usar adecuadamente un lenguaje de modelado (UML) para el análisis y diseño de sistemas 	
COMPETENCIAS DE FORMACIÓN:	
Competencias que compromete la asignatura:	Contribuye principalmente a las competencias del perfil: " <i>Desarrolla Habilidades en el uso de las tecnologías de la información y de la comunicación para la gestión de datos espaciales</i> " y " <i>Demuestra capacidades para asimilar, aplicar, investigar y desarrollar nuevas tecnologías para la gestión de la información espacial</i> ", en donde se debe analizar e implementar un programa aplicando el paradigma de programación orientado a objetos
Subcompetencias de la asignatura:	<ul style="list-style-type: none"> Comprender el concepto de paradigma y sus implicaciones en el modo de resolver problemas. Identificar el tipo de problemas en los cuales se aplique adecuadamente el paradigma orientado a objetos. Contextualizar el modo en que el paradigma orientado a objetos ayuda a mejorar las capacidades de reutilización del software. Comprende los conceptos de clase, atributo, operación, interfaz y objeto. Aplicar el mecanismo de paso de mensajes entre clases. Identificar y ser capaz de implementar los distintos tipos de relaciones que se pueden establecer a nivel de objeto entre clases: asociaciones, agregaciones y composiciones. Comprende el proceso de simplificación de la realidad a través de un lenguaje de modelado. Entender la relación entre diagramas de clase y código de implementación de dichos diagramas.
Programa sintético:	<ol style="list-style-type: none"> Introducción al paradigma Orientado a Objetos <ol style="list-style-type: none"> El progreso de la abstracción El paradigma orientado a objetos Lenguajes orientados a objetos Metas del paradigma orientado a objetos

	<ul style="list-style-type: none">2. Fundamentos de la programación orientada a objetos<ul style="list-style-type: none">2.1. Clases2.2. Atributos2.3. Operaciones (métodos)2.4. El concepto de interfaz2.5. El concepto de objeto2.6. Metaclases2.7. El diseño de aplicaciones OO2.8. Relaciones entre clases y relaciones entre objetos2.9. Documentación del código2.10. Constructores3. HERENCIA<ul style="list-style-type: none">3.1. Introducción a la Herencia3.2. Herencia Simple4. POLIMORFISMO<ul style="list-style-type: none">4.1. Polimorfismo y reutilización4.2. Sobrecarga4.3. Polimorfismo en jerarquías de herencia4.4. Variables Polimórficas4.5. Persistencia
--	---

III. ESTRATEGIAS

Metodología Pedagógica y Didáctica:

- Asistencia a clases expositivas y de discusión
- Elaboración y lectura de paper (documentación).
- Se debe procurar incentivar el trabajo de grupo más que el trabajo individual. (se recomienda trabajar en grupos de dos o tres estudiantes)
- Implementación y prueba de prototipos (programas) en laboratorio de computación

	Horas			Horas profesor/semana	Horas Estudiante/semana	Total Horas Estudiante/semestre	Créditos
Tipo de Curso	TD	TC	TA	(TD + TC)	(TD + TC +TA)	X 16 semanas	
Asignatura	2	4	3	6	9	144	3

Trabajo Presencial Directo (TD): trabajo de aula con plenaria de todos los estudiantes.

Trabajo Mediado_Cooperativo (TC): Trabajo de tutoría del docente a pequeños grupos o de forma individual a los estudiantes.

Trabajo Autónomo (TA): Trabajo del estudiante sin presencia del docente, que se puede realizar en distintas instancias: en grupos de trabajo o en forma individual, en casa o en biblioteca, laboratorio, etc.)

IV. RECURSOS

Medios y Ayudas:

- Aula normal con pizarrón para sesiones de cátedra y para sesiones de discusión.
- Disponibilidad para acceder a proyector multimedia.
- Laboratorio de computación, para las sesiones de laboratorio.
- IDE's para desarrollar en java (Eclipse, Netbeans, ...)
- Página web para publicar material didáctico, guías de ejercicios, soluciones, tareas, etc.
- Acceso al material bibliográfico recomendado.
- Asignación de una persona que tenga las plenas competencias del curso (monitor) para asesorar a los estudiantes en dudas durante las sesiones del laboratorio de computación.

BIBLIOGRAFÍA

TEXTOS GUÍA

- Joyanes, Luís. Programación Orientada por Objetos.
- Deitel & Deitel. C++ Cómo programar. Ed. Prentice Hall.
- Booch, Grady et al. El Lenguaje Unificado de Modelado. Ed. Addison-Wesley.

TEXTOS COMPLEMENTARIOS

REVISTAS

DIRECCIONES DE INTERNET

V. ORGANIZACIÓN / TIEMPOS

Espacios, Tiempos, Agrupamientos:

Se recomienda trabajar una unidad cada cuatro semanas, trabajar en pequeños grupos de estudiantes, utilizar Internet para comunicarse con los estudiantes para revisiones de avances y solución de preguntas (esto considerarlo entre las horas de trabajo cooperativo).

VI. EVALUACIÓN

PRIMERA NOTA	TIPO DE EVALUACIÓN	FECHA	PORCENTAJE
	Prueba oral/escrita para el grupo que el docente elabora.	Semana 5 ó 6	15 %
SEGUNDA NOTA	Prueba escrita conjunta (para todos los grupos de la asignatura), elaborada por todos los docentes que imparten la asignatura.	Semana 14 ó 15	20 %
TERCERA NOTA	Guías de ejercicios resueltas Informes de conceptos basado en análisis (Paper's) Pruebas orales/escritas rápidas (Quizes)	Varias fechas	10 %
CUARTA NOTA	Informe de desempeño en laboratorio (Para 3 o 4)	Varias fechas	25 %
PROYECTO FINAL	Informe de desempeño y sustentación de un prototipo funcional que evalúe las competencias exigidas.	Semana 17 - 18	30 %

ASPECTOS A EVALUAR DEL CURSO

- Claridad y entendimiento de los conceptos.
- Que se haya identificado correctamente el problema y que el modelo lo represente adecuadamente.
- Que la solución diseñada resuelva el problema.
- Apego a la formalidad y estándares requeridos.
- Que el análisis de corrección sea exhaustivo.
- Que el prototipo corresponda al modelo diseñado y no presente errores de sintaxis.
- La asistencia a las clases magistrales y a los laboratorios.
- El esfuerzo y dedicación en la resolución de problemas.
- Que la documentación permita reconocer la forma en que se ha abordado el problema y la estructura del programa implementado.
- En las pruebas escritas se consideran en forma parcial los aspectos considerados en proyectos de programación bajo problemas que requieren un menor tiempo de desarrollo y en una modalidad que no requiere uso del computador, así como la comprensión conceptual.

DATOS DEL DOCENTE			
NOMBRE :			
PREGRADO :			
POSTGRADO :			
ASESORIAS: FIRMA DE ESTUDIANTES			
NOMBRE	FIRMA	CÓDIGO	FECHA
1.			
2.			
3.			
FIRMA DEL DOCENTE			
<div></div>			
FECHA DE ENTREGA: _____			