

1. Datos Generales de la asignatura

Nombre de la asignatura:	Programación Avanzada
Clave de la asignatura:	MTG-1023
SATCA¹:	3 – 3 – 6
Carrera:	Ingeniería Mecatrónica

2. Presentación

Caracterización de la asignatura

Esta asignatura aporta al perfil del Ingeniero Mecatrónico la capacidad de análisis, desarrollo e implementación de software de aplicación orientado a objetos y visual cumpliendo con estándares de calidad, con el fin de apoyar la productividad y competitividad de los sistemas mecatrónicos.

Proporciona al estudiante las competencias necesarias para abordar el estudio de cualquier lenguaje orientado a objetos y visual que es de gran utilidad para resolver problemas de productividad computacional.

Es también base fundamental del perfil del egresado y relacionada con todas aquellas en la que el pensamiento lógico es requerido.

Intención didáctica

El enfoque sugerido para la materia requiere que las actividades prácticas promuevan el desarrollo de habilidades para la resolución de problemas, tales como: identificación, manejo, control de variables, datos relevantes, planteamiento de hipótesis, trabajo en equipo, asimismo, propicien procesos intelectuales como inducción-deducción y análisis-síntesis con la intención de generar una actividad intelectual compleja; las actividades teóricas se han descrito como actividades previas al tratamiento práctico de los temas. En las actividades prácticas sugeridas, es conveniente que el profesor sólo guíe al estudiante en la construcción de su conocimiento.

El primer tema se presenta los conceptos de la programación orientada a objetos y visual, teniendo la intención de introducir al estudiante en los elementos del modelo de objetos, así como el uso básico del lenguaje de modelado unificado.

El segundo tema se centra en la definición e implementación de clases y objetos permitiendo al estudiante adquirir las competencias fundamentales de la programación orientada a objetos.

El tercer tema tiene como propósito la creación de objetos que incorporen propiedades y métodos de otros objetos (herencia), construyéndolos a partir de éstos sin necesidad de reescribirlo todo.

El cuarto tema trata una de las características fundamentales de la programación orientada a objetos; el polimorfismo, que permite reutilizar métodos con el mismo nombre, pero con relación a la clase a la que pertenece cada uno, con comportamientos diferentes. Además, el estudiante adquirirá los

¹ Sistema de Asignación y Transferencia de Créditos Académicos

conocimientos para tratar situaciones excepcionales que se presentan en tiempo de ejecución.

El tema cinco aborda el tema de la programación con controles y componentes, acceso a los datos, ambiente integrado de desarrollo y los proyectos que se pueden generar ahí.

En el tema seis se presentan las formas, los controles, cajas de diálogos y el manejo de eventos necesarios para crear una interfaz gráfica de usuario

Se sugiere que se valla generando a lo largo del curso un programa que integre todos los temas vistos, con el objeto de reforzar los aprendizajes obtenidos. En esta actividad, se propone la programación de un sistema que, de solución a una problemática real, buscando que el estudiante tenga contacto con los conceptos en forma concreta y sea a través del análisis, creatividad e imaginación que se logre tal objetivo; El programa integrador se propone que se defina en el transcurso del primer tema de esta competencia.

En el transcurso de las actividades programadas es muy importante que el estudiante aprenda a valorar las actividades que lleva a cabo y entienda que está construyendo su hacer futuro y en consecuencia actúe de una manera profesional; de igual manera, aprecie la importancia del conocimiento y los hábitos de trabajo; desarrolle la capacidad de análisis, precisión y la curiosidad, la puntualidad, el entusiasmo y el interés, la tenacidad, la flexibilidad y la autonomía.

Es necesario que el profesor ponga atención y cuidado en estos aspectos en el desarrollo de las actividades de aprendizaje de esta asignatura.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Evento
Instituto Tecnológico Superior de Irapuato, del 24 al 28 de agosto de 2009.	Representantes de los Institutos Tecnológicos de: Apizaco, Celaya, Ciudad Cuauhtémoc, Cuautla, Durango, Guanajuato, Hermosillo, Huichapan, Irapuato, Jilotepec, Jocotitlán, La Laguna, Oriente del Estado de Hidalgo, Pabellón de Arteaga, Parral, Reynosa, Saltillo, San Luis Potosí, Tlalnepantla, Toluca y Zacapoaxtla.	Reunión Nacional de Diseño e Innovación Curricular para el Desarrollo y Formación de Competencias Profesionales de las Carreras de Ingeniería Eléctrica, Ingeniería Electromecánica, Ingeniería Electrónica e Ingeniería Mecatrónica.
Instituto Tecnológico de Mexicali, del 25 al 29 de enero del 2010.	Representantes de los Institutos Tecnológicos de: Apizaco, Celaya, Ciudad Cuauhtémoc, Cuautla, Durango, Guanajuato, Hermosillo, Huichapan, Irapuato, Jilotepec, Jocotitlán, La Laguna, Mexicali,	Reunión Nacional de Consolidación de los Programas en Competencias Profesionales de las Carreras de Ingeniería Eléctrica, Ingeniería Electromecánica, Ingeniería Electrónica e Ingeniería

	Oriente del Estado de Hidalgo, Pabellón de Arteaga, Reynosa, Saltillo, San Luis Potosí, Toluca y Zacapoaxtla.	Mecatrónica.
Instituto Tecnológico de la Laguna, del 26 al 29 de noviembre de 2012.	Representantes de los Institutos Tecnológicos de: Apizaco, Celaya, Chapala, Cd. Cuauhtémoc, Colima, Culiacán, Huixquilucan, La Laguna, León, Nuevo Laredo, Nuevo León, Oriente del Estado de Hidalgo, Querétaro, Tlalnepantla, Uruapan, Veracruz y Zacapoaxtla.	Reunión Nacional de Seguimiento Curricular de los Programas en Competencias Profesionales de las Carreras de Ingeniería Eléctrica, Ingeniería Electromecánica, Ingeniería Electrónica, Ingeniería Mecánica e Ingeniería Mecatrónica.
Instituto Tecnológico de Toluca, del 10 al 13 de febrero de 2014.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Boca del Río, Celaya, Mérida, Orizaba, Puerto Vallarta y Veracruz.	Reunión de Seguimiento Curricular de los Programas Educativos de Ingenierías, Licenciaturas y Asignaturas Comunes del SNIT.
Tecnológico Nacional de México, del 25 al 26 de agosto de 2014.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Apizaco, Boca del Río, Celaya, Cerro Azul, Cd. Juárez, Cd. Madero, Chihuahua, Coacalco, Coatzacoalcos, Durango, Ecatepec, La Laguna, Lerdo, Matamoros, Mérida, Mexicali, Motúl, Nuevo Laredo, Orizaba, Pachuca, Poza Rica, Progreso, Reynosa, Saltillo, Santiago Papasquiario, Tantoyuca, Tlalnepantla, Toluca, Veracruz, Villahermosa, Zacatecas y Zacatepec. Representantes de Petróleos Mexicanos (PEMEX).	Reunión de trabajo para la actualización de los planes de estudio del sector energético, con la participación de PEMEX.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Aplica tecnologías y herramientas actuales y emergentes para desarrollar sistemas de información.

5. Competencias previas

<ul style="list-style-type: none"> • Aplica técnicas de modelado para la solución de problemas. • Aplica la sintaxis de un lenguaje de programación estructurado. • Aplica un lenguaje estructurado para la solución de problemas. • Aplica las estructuras de datos en la elaboración de programas.
--

6. Temario

No.	Nombre de temas	Subtemas
1	Introducción	<p>1.1 Paradigma de la POO y visual</p> <p>1.2 Lenguajes de programación visual y orientada a eventos</p> <p>1.3 Elementos del modelo de objetos: clases, objetos, abstracción, modularidad, encapsulamiento, herencia y polimorfismo</p>
2	Objetos y Clases	<p>2.1 Declaración de clases: atributos, métodos, encapsulamiento.</p> <p>2.2 Instanciación de una clase.</p> <p>2.3 Referencia al objeto actual.</p> <p>2.4 Métodos: declaración, mensajes, paso de parámetros, retorno de valores.</p> <p>2.5 Constructores y destructores: declaración, uso y aplicaciones.</p> <p>2.6 Sobrecarga de métodos.</p> <p>2.7 Sobrecarga de operadores</p>
3	Herencia	<p>3.1 Definición: clase base, clase derivada.</p> <p>3.2 Clases base públicas, protegidas y privadas.</p> <p>3.3 Referencia al objeto de la clase base.</p> <p>3.4 Constructores y destructores en clases derivadas.</p>
4	Polimorfismo y Excepciones	<p>4.1 Definición.</p> <p>4.2 Clases abstractas: definición, métodos abstractos, implementación de clases abstractas, modelado de clases abstractas.</p> <p>4.3 Reutilización de código.</p> <p>4.4 Tipos de excepciones.</p> <p>4.5 Gestión de excepciones: manejo de excepciones, lanzamiento de excepciones.</p>
5	Programación Visual	<p>5.2 Objetos, controles y componentes.</p> <p>5.3 Tecnología .NET.</p> <p>5.4 Entorno integrado de desarrollo.</p> <p>5.5 Tipos de proyectos.</p>
6	Formas, Controles y Eventos	<p>6.1 Controles estándar</p> <p>6.2 Eventos y propiedades del formulario.</p> <p>6.3 Tipos de formulario.</p> <p>6.4 Control de Eventos</p> <p>6.5 Cajas de diálogo.</p>

	6.6 Menú. 6.7 Diseño de Interfaz Grafica de Usuario
--	--

7. Actividades de aprendizaje de los temas

1 Introducción	
Competencias	Actividades de aprendizaje
<p>Específica(s): Comprende, describe y modela los conceptos principales del paradigma de programación orientado a objetos y visual y los aplica a situaciones de la vida real.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis • Capacidad de organización y planificación • Habilidad para la búsqueda y análisis de información proveniente de fuentes diversas. • Trabajo en equipo 	<ul style="list-style-type: none"> • Investigar y seleccionar en diversas fuentes de información los conceptos principales del paradigma de programación orientado a objetos y visual. • Identificar ejemplos de la vida real que apliquen o manifiesten dichos conceptos. • Redactar una definición propia de los conceptos de forma simple y entendible. • Comentar en clase las definiciones de otros compañeros para enriquecer la propia y consensar una grupal. • Desarrollar un mapa conceptual entre los distintos paradigmas señalando sus ventajas y desventajas.
2 Objetos y Clases	
Competencias	Actividades de aprendizaje
<p>Específica(s): Implementa clases y objetos cumpliendo las reglas de la programación orientada a objetos.</p> <p>Implementa constructores y destructores para inicializar atributos y liberar recursos.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis • Capacidad de organización y planificación. • Conocimientos básicos de la carrera • Habilidades básicas de manejo de la computadora • Habilidad de manejo de software de Ingeniería. • Habilidad para la búsqueda y análisis de información proveniente de fuentes diversas • Solución de problemas • Capacidad de aplicar los conocimientos en la práctica • Habilidades de investigación 	<ul style="list-style-type: none"> • Realizar programas de clases con atributos públicos para exponer y comprender la vulnerabilidad de los datos. • Proteger los atributos con modificadores de acceso privados o protegidos y programa métodos públicos para otorgar acceso seguro a los mismos. • Reunir dentro de una clase los miembros necesarios para resolver un problema en particular, y así implementa el encapsulamiento. • Instanciar objetos para identificar el nacimiento y muerte de los mismos. • Programar constructores y destructores para las clases, de manera que permitan dar un valor inicial a sus atributos cuando nazcan sus objetos, o liberar recursos cuando mueran los mismos. • Identificar los comportamientos de una clase que pueden variar dependiendo del paso, cantidad, tipo u orden de

	<p>argumentos. Programa cada variación del comportamiento en métodos sobrecargados para agregar flexibilidad a la clase.</p> <ul style="list-style-type: none"> • Identificar operaciones que puedan ser realizadas entre dos objetos de la misma clase. • Seleccionar un operador existente del lenguaje y sobrecargarlo en la clase de los objetos para implementarles dicha funcionalidad.
3 Herencia	
Competencias	Actividades de aprendizaje
<p>Específica(s): Implementa la herencia en clases derivadas para reutilizar los miembros de una clase base.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis • Capacidad de organización y planificación. • Conocimientos básicos de la carrera • Habilidades básicas de manejo de la computadora • Habilidad de manejo de software de Ingeniería • Habilidad para la búsqueda y análisis de información proveniente de fuentes diversas. • Solución de problemas • Habilidad de modelar • Capacidad de aplicación de los conocimientos en la práctica • Habilidades de investigación 	<ul style="list-style-type: none"> • Analizar analogías taxonómicas de los seres vivos que compartan rasgos comunes por estar relacionados mediante una herencia genética e identifica la especie a la que pertenecen. • Identificar los atributos y comportamientos propios de una especie que comparten los animales pertenecientes a ella. • Analizar objetos reales que compartan características comunes por pertenecer a una misma categoría de objetos. • Identificar los atributos y comportamientos propios de una categoría de objetos que compartan todos sus miembros. • Investigar en fuentes de información los conceptos relacionados con la herencia y su implementación en un lenguaje de programación orientado a objetos. • Programar una clase base para una especie de animales con los atributos y comportamientos comunes a todos los animales pertenecientes a ella. • Implementar clases derivadas para animales pertenecientes a la misma especie de la cual se programó la clase base anteriormente. • Especializar cada clase derivada con comportamientos y atributos específicos de un tipo de animal para identificarlo y distinguirlo de los demás.

	<ul style="list-style-type: none"> • Crear varias instancias de clases derivadas diferentes para verificar la existencia de los miembros heredados comunes en todas ellas, y la diversidad de sus especializaciones. • Repetir las mismas actividades, pero utilizando objetos y categorías de objetos reales. • Sobrecargar los constructores de las clases base y derivadas para analizar y experimentar el comportamiento y uso de los constructores en combinación con la herencia. • Analizar qué animales u objetos de la vida real rompen algún comportamiento heredado para reinventar el suyo propio por sobre el resto de sus parientes que siguen respetando el heredado.
4 Polimorfismo y Excepciones	
Competencias	Actividades de aprendizaje
<p>Específica(s): Implementa interfaces y clases polimórficas.</p> <p>Sobrecarga métodos y operadores para la optimización del código de una clase</p> <p>Identifica, maneja, gestiona y crea las condiciones de error que interrumpan el flujo normal de ejecución de un programa.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis • Capacidad de organización y planificación. • Conocimientos básicos de la carrera • Habilidades básicas de manejo de la computadora. • Habilidad de manejo de software de Ingeniería. • Habilidad para la búsqueda y análisis de información proveniente de fuentes diversas. • Solución de problemas • Habilidad de modelar • Capacidad de aplicación de los conocimientos 	<ul style="list-style-type: none"> • Analizar clases base que no requieran ser instanciadas, o que carezcan de sentido para ello por ser abstractas. • Investigar en fuentes de información los conceptos y reglas para implementar clases abstractas en un lenguaje de programación orientado a objetos. • Implementar clases abstractas en clases base que no requieran ser instanciadas con al menos un método abstracto para que sea implementado por sus clases derivadas en múltiples formas. • Implementar una clase con todos sus comportamientos abstractos. Investigar en diversas fuentes de información el concepto de interfaz y compararlo con la clase cien por ciento abstracta. • Programar interfaces para definir los comportamientos que una clase deberá de tener al implementarla. • Implementar una misma interfaz en diferentes clases para dar en cada una un comportamiento diferente a sus métodos.

<p>en la práctica</p> <ul style="list-style-type: none"> Habilidades de investigación 	
<p>Tema 5: Programación Visual</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s): Manipula los controles y componentes estándar definidos en el entorno de desarrollo y el lenguaje para el desarrollo de aplicaciones.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> Capacidad de análisis y síntesis Capacidad de organizar y planificar Habilidades básicas de manejo de la computadora Habilidad de manejo de software de Ingeniería Habilidad para buscar y analizar información proveniente de fuentes diversas Solución de problemas Capacidad de aplicar los conocimientos en la práctica Habilidades de investigación 	<ul style="list-style-type: none"> Exploración del entorno integrado de desarrollo del lenguaje de programación seleccionado, lo cual permita identificar los métodos, eventos, controles y componentes estándar. Demostrar con ejemplos que utilicen los principales controles y componentes estándar, así como los métodos y eventos.
<p>Tema 6: Formas, Controles y Eventos</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s): Manipula los controles y componentes estándar definidos en el entorno de desarrollo y el lenguaje para el desarrollo de aplicaciones.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> Capacidad de análisis y síntesis Capacidad de organizar y planificar Habilidades básicas de manejo de la computadora Habilidad de manejo de software de Ingeniería Habilidad para buscar y analizar información proveniente de fuentes diversas Solución de problemas Capacidad de aplicar los conocimientos en la práctica Habilidades de investigación 	<ul style="list-style-type: none"> Demostrar con ejemplos que utilicen los principales controles y componentes estándar, así como los métodos y eventos. Realizar un caso de estudio que presente una problemática real en la cual se solicite una propuesta de solución GUI mediante la aplicación de métodos, eventos, controles y componentes estándar

8. Práctica(s)

- Crea un programa que instancie y use un objeto predefinido por el lenguaje para practicar el envío de mensajes, el uso de parámetros y la recepción de su respuesta. Sugerencia: objeto de clase String.
- Analiza objetos concretos (puerta, elevador, televisor, etc.) y abstractos (cuenta bancaria, préstamo, viaje, etc.) de la vida real para abstraer y modelar sus atributos y comportamientos. Implementa clases para instanciar objetos que modelen sus contrapartes de la vida real usando tipos de datos simples y objetos como parámetros y valores de retorno, así como métodos sin valores de retorno.
- Intercambia clases de objetos entre compañeros para usar sus miembros con valores o situaciones erróneas que evidencien la necesidad de protegerlos con modificadores de acceso. Modifica el código fuente aplicando los distintos niveles de acceso para experimentar y descubrir (aprender) el impacto de cada uno de ellos.
- Implementa la clase Persona con los atributos nombre y edad; un constructor, un destructor, y al menos el método crecer para mapear el ciclo de vida de una persona con el de un objeto.
- Implementa la clase Calculadora que realice al menos las cuatro operaciones básicas de la aritmética sobrecargando métodos para cada tipo de dato numérico del lenguaje de los parámetros.
- Implementa la clase Matriz que sobrecargue los operadores +, -, * y / para este tipo de dato definido por el usuario.
- Programa una aplicación sobre figuras geométricas que implemente la clase base FiguraGeometrica de la cual hereden sus miembros las clases derivadas y que éstas solo especialicen sus características o comportamientos.
- Implementa constructores y destructores a las clases base y derivadas de la aplicación sobre figuras geométricas para experimentar y comprender su funcionamiento cuando está implicada la herencia.
- Modifica la clase FiguraGeometrica para convertirla en abstracta y programar al menos un método abstracto que todas las clases derivadas deberán implementar con su propio comportamiento.
- Programa la interfaz Vehículo con un conjunto de métodos abstractos que todo vehículo de la vida real debería tener. Programa varias clases que implementen la interfaz anterior y definan el comportamiento particular de sus métodos.
- Programa clases que generen excepciones comunes como referencias nulas o desbordamientos numéricos para estudiar su naturaleza, comportamiento, prevención y lanzamiento.
- Implementar aplicaciones que almacenen y recuperen información de diferentes tipos de datos simples a través de un archivo de texto para persistir información.
- Desarrollar una aplicación demostrativa sobre el uso de diversos controles y componentes estándar, programando para cada control o componente algunos de sus eventos. Algunos de los controles y componentes a considerar:
 - Formularios
 - Cuadros de textos
 - Etiquetas
 - Listas de Selección
 - Cuadros de verificación
 - Botones
 - Menús
 - Ventanas modales y no modales
 - Manejo de contenido enriquecido (video, sonido, animación, etc.).

9. Proyecto de asignatura

El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

- **Fundamentación:** marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.
- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

10. Evaluación por competencias

Son las técnicas, instrumentos y herramientas sugeridas para constatar los desempeños académicos de las actividades de aprendizaje.

- Lista de cotejo o rubrica, por tema.
- Guías de observación.
- Portafolio de evidencias.
- Exposiciones orales.
- Proyectos.
- Exámenes.

11. Fuentes de información

1. Taylor David. Object Orient informations systems, planning and implementations. Canada: Wiley. 1992.
2. Larman Craig. UML y patrones introducción al análisis y diseño orientado a objetos. México: Prentice Hall. 1999.
3. Winblad, Ann L. Edwards, Samuel R. Software orientado a objetos. USA: Addison. Wesley/ Díaz Santos. 1993.
4. Fco. Javier Ceballos. Java 2 Curso de Programación. Alfaomega.
5. Agustín Froufe. Java 2 Manual de usuario y tutorial. Alfaomega.
6. Laura Lemay, Rogers Cadenhead. Aprendiendo JAVA 2 en 21 días. Prentice Hall.
7. Herbert Schildt. Fundamentos de Programación en Java 2. McGrawHil.
8. J Deitel y Deitel. Como programar en Java. Prentice Hall.
9. Stephen R. Davis. Aprenda Java Ya. McGrawHill.
10. Kris Jamsa Ph D. ¡Java Ahora!. McGrawHill.
11. Francisco Charte Ojeda. Visual C# .NET. ANAYA MULTIMEDIA
12. Kingsley-Hughes, Kathie; Kingsley-Hughes, Adrian. C# 2005. ANAYAMULTIMEDIA

13. Ceballos Francisco Javier. Enciclopedia de Microsoft Visual C#. 2ª Edición
14. El lenguaje de programación C#. Fco. Javier Ceballos Sierra. Editorial Ra-ma.
15. Tom Archer. A fondo C#. McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S.A.U.
16. Firtman, Maximiliano R. Visual Studio .Net Framework 3.5 para Profesionales. 1ra. edición. Ed. Alfaomega Grupo Editor. Argentina. 2008. 372 p.
17. Joyanes, Luis. Programación en C/C++Java y UML. 1ra. edición. Ed. McGraw Hill Interamericana. México. 2009. 880 p.
18. Ramírez, Felipe. Introducción a la Programación, Algoritmos y su Implementación en VB.Net, C#, Java y C++. 2da. Edición. Ed. Alfa-Omega Grupo Editor. México. 2007. 520 p.
19. Sharp, John. Visual C# 2008. 1ra. edición. Ed. Anaya Multimedia. España. 2008. 832 p.