

UNIVERSIDAD DE GUANAJUATO										
NOMBRE DE LA ENTIDAD:		CAMPUS LEÓN; DIVISIÓN DE CIENCIAS E INGENIERÍAS								
NOMBRE DEL PROGRAMA EDUCATIVO:		Licenciatura en Ingeniería Biomédica								
NOMBRE DE LA MATERIA:		Programación orientada a objetos y eventos					CLAVE:		BMCPOOE-03	
FECHA DE ELABORACIÓN:		30 de mayo de 2011					HORAS/SEMANA/SEMESTRE			
FECHA DE ACTUALIZACIÓN:										
		Dr. Arturo González Vega					TEORÍA:		2	
PRERREQUISITOS:										
CURSADA Y APROBADA:		Ninguno					PRÁCTICA:		3	
CURSADA:		Ninguno					CRÉDITOS:		7	
CARACTERIZACIÓN DE LA MATERIA										
POR EL TIPO DE CONOCIMIENTO:		DISCIPLINARIA		FORMATIVA	X	METODOLÓGICA				
POR LA DIMENSIÓN DEL CONOCIMIENTO:		ÁREA BÁSICA	X	ÁREA GENERAL		ÁREA PROFESIONAL				
POR LA MODALIDAD DE ABORDAR EL CONOCIMIENTO:		CURSO	X	TALLER		LABORATORIO		SEMINARIO		
POR EL CARÁCTER DE LA MATERIA:		OBLIGATORIA	X	RECURSABLE		OPTATIVA		SELECTIVA		
ES PARTE DE UN TRONCO COMÚN O MATERIAS COMUNES:		SÍ	X	NO				ACREDITABLE		
COMPETENCIA (S) GENERAL(ES) DE LA MATERIA:										
<p>Se desarrollará el paradigma de programación orientada a objetos. Se basará el paradigma de programación en el lenguaje C++. Comprenderá la programación orientada a eventos y se harán aplicaciones de GUI en aplicaciones basadas en C++ (Visual C++, QT). Se comprenderá el concepto de multihilos (multithreds), con miras a su aplicación en programación en paralelo.</p>										
CONTRIBUCIÓN DE LA MATERIA AL LOGRO DEL PERFIL POR COMPETENCIAS.										
<p>La materia de programación orientada a objetos y eventos contribuye a las competencias específicas metodológicas de la siguiente manera: C3. Demuestra una comprensión de los conceptos básicos y principios fundamentales del área Ingeniería en medicina. M1. Plantea, analiza y resuelve problemas de Ingeniería Biomédica, tanto teóricos como experimentales, mediante la utilización de métodos analíticos, experimentales o numéricos. M3. Verifica y evalúa el ajuste de modelos a la realidad, identificando su dominio de validez. M8. Utiliza y elabora programas o sistemas embebidos (hardware y software) para el procesamiento de información, cálculo numérico, simulación y/o control de procesos biomédicos, así como diseño y desarrollo de experimentos biomédicos. M9. Diseña, desarrolla y utiliza tecnología para el procesamiento de información, cálculo numérico, simulación de procesos biomédicos y/o control de experimentos. M10. Analiza y verifica tecnología para el procesamiento, adquisición y transmisión de información, cálculo numérico, simulación de procesos biomédicos y/o control de experimentos en el área de la salud. LS1. Participa en actividades profesionales relacionadas con tecnologías de alto nivel, sea en el laboratorio o en la industria médica. LS2. Participa en asesorías y elaboración de propuestas de ciencia y tecnología en temas con impacto económico y social en el ámbito nacional.</p>										

PRESENTACIÓN DE LA MATERIA

Esta materia tiene como objetivo el aprendizaje del paradigma de programación orientada a objetos. El aprendizaje de dicho paradigma tiene como objetivo el uso de esta forma de programación para la generación de GUI's tanto en plataformas de programación comerciales (como Visual C++) como plataformas de software libre como QT. Otro punto muy importante es el conocimiento y aplicación de los hilos y multihilos que permitirán al estudiante prepararse teóricamente para abordar el uso de la programación en paralelo basado en procesadores gráficos.

RELACIÓN CON OTRAS MATERIAS DEL PLAN DE ESTUDIOS

Se recomienda fuertemente haber cursado la materia de Programación básica.

Esta materia está diseñada para dar las herramientas formales diseñar y programar un algoritmo que permita la solución de programas generales con el paradigma orientado a objetos.

Esta materia proveerá los insumos para poder generar programas de cómputo, tanto en C para poder abordar problemas donde el uso de computadoras dé solución a problemas que pueden llegar a ser complejos con interfaces de usuario profesionales.

NOMBRE DE LA UNIDAD TEMÁTICA/BLOQUE TEMÁTICO:		Paradigma de la programación orientada a objetos			TIEMPO ESTIMADO PARA DESARROLLAR LA UNIDAD TEMÁTICA:	10 horas (2 semanas)			
COMPETENCIAS A DESARROLLAR		SABERES			EVIDENCIAS DE DESEMPEÑO				
		CONOCIMIENTOS	HABILIDADES	ACTITUDES	DIRECTA	POR PRODUCTO			
1. Comprender el paradigma orientado a objetos.	2. Entender y manejar el concepto de abstracción de datos.	3. Reconocer las diferencias básicas entre C y C++	4. Aprender las herramientas básicas de C++	5. Analizar,	<ul style="list-style-type: none"> o Paradigma orientado a objetos. o Abstracción de datos o Compilador C++. o Clase en el contexto de programación orientada a objetos. 	<ul style="list-style-type: none"> - Comprender el paradigma orientado a objetos. - Entender y manejar el concepto de abstracción de datos. - Reconocer las diferencias básicas entre C y C++ - Aprender las herramientas básicas de C++ - Analizar, 	<ul style="list-style-type: none"> • El fortalecimiento de hábitos correctos de estudio y análisis. • La valoración de la actividad creadora y la imaginación • La ética profesional al no falsificar información • La aceptación de los alcances y las limitaciones personales. • Proponer estrategias para la solución de problemas. • La valoración de la investigación interdisciplinaria y 	<ol style="list-style-type: none"> 1. Participación en clase 2. Ejercicios en pizarrón 3. Participación grupal en sesiones de ejercicios. 4. Evaluación mediante exámenes sorpresa de corta duración que cuenten como tarea. 5. Elaboración de prácticas de laboratorio 	<ul style="list-style-type: none"> Tareas Exámenes Sorpresa Examen sumativo Cuaderno de ejercicios. Reporte de prácticas de laboratorio

comprender y aplicar el concepto de clase.		comprender y aplicar el concepto de clase.	multidisciplinaria • La valoración de la actividad creadora y la imaginación		
--	--	--	---	--	--

NOMBRE DE LA UNIDAD TEMÁTICA/BLOQUE TEMÁTICO:	Clases	TIEMPO ESTIMADO PARA DESARROLLAR LA UNIDAD TEMÁTICA:	5 horas (1 semanas)
--	--------	---	------------------------

COMPETENCIAS A DESARROLLAR	SABERES			EVIDENCIAS DE DESEMPEÑO	
	CONOCIMIENTOS	HABILIDADES	ACTITUDES	DIRECTA	POR PRODUCTO
1. Profundizar en el concepto de clase. 2. Comprender las diferencias entre datos y funciones privados, públicos y reservados. 3. Comprender el uso y la sintaxis para la definición de Constructores y destructor de una clase.	<ul style="list-style-type: none"> ○ Clase en programación orientada a objetos ○ Tipos de datos: privados, públicos y reservados. ○ Constructores de una clase. ○ Destructor de una clase. 	<ul style="list-style-type: none"> - Profundizar en el concepto de clase. - Comprender las diferencias entre datos y funciones privados, públicos y reservados. - Comprender el uso y la sintaxis para la definición de Constructores y destructor de una clase. 	<ul style="list-style-type: none"> • El fortalecimiento de hábitos correctos de estudio y análisis. • La valoración de la actividad creadora y la imaginación • La ética profesional al no falsificar información • La aceptación de los alcances y las limitaciones personales. • Proponer estrategias para la solución de problemas. • La valoración de la investigación interdisciplinaria y multidisciplinaria • La valoración de la actividad creadora y la imaginación 	1. Participación en clase 2. Ejercicios en pizarrón 3. Participación grupal en sesiones de ejercicios. 4. Evaluación mediante exámenes sorpresa de corta duración que cuenten como tarea. 5. Elaboración de prácticas de laboratorio	Tareas Exámenes Sorpresa Examen sumativo Cuaderno de ejercicios. Reporte de prácticas de laboratorio

NOMBRE DE LA UNIDAD TEMÁTICA/BLOQUE TEMÁTICO:	Conceptos de programación orientada a objetos.	TIEMPO ESTIMADO PARA DESARROLLAR LA UNIDAD TEMÁTICA:	15 horas (3 semanas)
--	--	---	-------------------------

COMPETENCIAS A	SABERES	EVIDENCIAS DE DESEMPEÑO
-----------------------	----------------	--------------------------------

DESARROLLAR	CONOCIMIENTOS	HABILIDADES	ACTITUDES	DIRECTA	POR PRODUCTO
1. Conocer, comprender y manipular el concepto de polimorfismo de funciones. 2. Conocer, comprender y manipular el concepto de herencia de clases. 3. Conocer, comprender y manipular el concepto de clases abstractas. 4. Conocer, comprender y manipular el concepto de ocultamiento de datos.	<ul style="list-style-type: none"> ○ Polimorfismo. ○ Herencia. ○ Clase abstracta. ○ Ocultamiento de datos. 	<ul style="list-style-type: none"> - Conocer, comprender y manipular el concepto de polimorfismo de funciones. - Conocer, comprender y manipular el concepto de herencia de clases. - Conocer, comprender y manipular el concepto de clases abstractas. - Conocer, comprender y manipular el concepto de ocultamiento de datos. 	<ul style="list-style-type: none"> • El fortalecimiento de hábitos correctos de estudio y análisis. • La valoración de la actividad creadora y la imaginación • La ética profesional al no falsificar información • La aceptación de los alcances y las limitaciones personales. • Proponer estrategias para la solución de problemas. • La valoración de la investigación interdisciplinaria y multidisciplinaria • La valoración de la actividad creadora y la imaginación 	1. Participación en clase 2. Ejercicios en pizarrón 3. Participación grupal en sesiones de ejercicios. 4. Evaluación mediante exámenes sorpresa de corta duración que cuenten como tarea. 5. Elaboración de prácticas de laboratorio	Tareas Exámenes Sorpresa Examen sumativo Cuaderno de ejercicios. Reporte de prácticas de laboratorio

NOMBRE DE LA UNIDAD TEMÁTICA/BLOQUE TEMÁTICO:	Control de nombres y pase por referencia	TIEMPO ESTIMADO PARA DESARROLLAR LA UNIDAD TEMÁTICA:	10 horas (2 semanas)
--	---	---	-----------------------------

COMPETENCIAS A DESARROLLAR	SABERES			EVIDENCIAS DE DESEMPEÑO	
	CONOCIMIENTOS	HABILIDADES	ACTITUDES	DIRECTA	POR PRODUCTO

<ol style="list-style-type: none"> 1. Comprender y usar las técnicas de control de nombres de funciones. 2. Comprender y utilizar el espacio de nombres. 3. Crear un espacio de nombres 4. Utilizar un espacio de nombres. 5. Comprender y utilizar el pase por referencia de datos en funciones. 6. Comprender la diferencia del uso de pase por referencia vs uso de apuntadores. 7. Entender el concepto de constructor por copia. 8. Conocer y manipular los apuntadores a funciones miembro. 	<ul style="list-style-type: none"> o Nombramiento de funciones. o Espacio de nombres o Paso por referencia de datos a una función. o Constructor por copia de una clase o Apuntadores a funciones miembro 	<ul style="list-style-type: none"> - Comprender y usar las técnicas de control de nombres de funciones. - Comprender y utilizar el espacio de nombres. - Crear un espacio de nombres - Utilizar un espacio de nombres. - Comprender y utilizar el pase por referencia de datos en funciones. - Comprender la diferencia del uso de pase por referencia vs uso de apuntadores. - Entender el concepto de constructor por copia. - Conocer y manipular los apuntadores a funciones miembro. 	<ul style="list-style-type: none"> • El fortalecimiento de hábitos correctos de estudio y análisis. • La valoración de la actividad creadora y la imaginación • La ética profesional al no falsificar información • La aceptación de los alcances y las limitaciones personales. • Proponer estrategias para la solución de problemas. • La valoración de la investigación interdisciplinaria y multidisciplinaria • La valoración de la actividad creadora y la imaginación 	<ol style="list-style-type: none"> 1. Participación en clase 2. Ejercicios en pizarrón 3. Participación grupal en sesiones de ejercicios. 4. Evaluación mediante exámenes sorpresa de corta duración que cuenten como tarea. 5. Elaboración de prácticas de laboratorio 	<p>Tareas Exámenes Sorpresa Examen sumativo Cuaderno de ejercicios. Reporte de prácticas de laboratorio</p>
---	--	---	---	--	---

NOMBRE DE LA UNIDAD TEMÁTICA/BLOQUE TEMÁTICO:	Sobrecarga de operadores, herencia y composición, funciones virtuales.	TIEMPO ESTIMADO PARA DESARROLLAR LA UNIDAD TEMÁTICA:	10 horas (2 semanas)
--	---	---	-------------------------

COMPETENCIAS A DESARROLLAR	SABERES			EVIDENCIAS DE DESEMPEÑO	
	CONOCIMIENTOS	HABILIDADES	ACTITUDES	DIRECTA	POR PRODUCTO
<ol style="list-style-type: none"> 1. Comprender y usar el concepto de sobrecarga de operadores. 2. Reconocer cuando 	<ul style="list-style-type: none"> o Sobrecarga de operadores. o Herencia en el contexto de programación 	<ul style="list-style-type: none"> - Comprender y usar el concepto de sobrecarga de operadores. - Reconocer cuando realizar una sobrecarga como 	<ul style="list-style-type: none"> • El fortalecimiento de hábitos correctos de estudio y análisis. • La valoración de la actividad creadora y la imaginación 	<ol style="list-style-type: none"> 1. Participación en clase 2. Ejercicios en pizarrón 3. Participación 	<p>Tareas Exámenes Sorpresa Examen sumativo Cuaderno de ejercicios.</p>

<p>realizar una sobrecarga como función amiga y cuando declarar la sobrecarga como función miembro.</p> <ol style="list-style-type: none"> 3. Comprender el concepto de herencia de clases. 4. Comprender el concepto de composición de clases. 5. Distinguir cuando es apropiado el uso de herencia y cuando se debe usar la composición de clases. 6. Entender y manipular el concepto de herencia múltiple. 7. Comprender el concepto de función virtual. 8. Conocer la sintaxis para la declaración de función virtual obligatoria. 9. Reconocer la forma de utilizar los Constructores virtuales. 10. Reconocer la forma de utilizar el Destructor virtual 	<p>orientada a objetos.</p> <ul style="list-style-type: none"> o Composición en el contexto de programación orientada a objetos. o Herencia múltiple o Funciones virtuales o Constructores virtuales o destructores virtuales. 	<p>función amiga y cuando declarar la sobrecarga como función miembro.</p> <ul style="list-style-type: none"> - Comprender el concepto de herencia de clases. - Comprender el concepto de composición de clases. - Distinguir cuando es apropiado el uso de herencia y cuando se debe usar la composición de clases. - Entender y manipular el concepto de herencia múltiple. - Comprender el concepto de función virtual. - Conocer la sintaxis para la declaración de función virtual obligatoria. - Reconocer la forma de utilizar los Constructores virtuales. - Reconocer la forma de utilizar el Destructor virtual 	<ul style="list-style-type: none"> • La ética profesional al no falsificar información • La aceptación de los alcances y las limitaciones personales. • Proponer estrategias para la solución de problemas. • La valoración de la investigación interdisciplinaria y multidisciplinaria • La valoración de la actividad creadora y la imaginación 	<p>grupal en sesiones de ejercicios.</p> <ol style="list-style-type: none"> 4. Evaluación mediante exámenes sorpresa de corta duración que cuenten como tarea. 5. Elaboración de prácticas de laboratorio 	<p>Reporte de prácticas de laboratorio</p>
---	---	---	--	---	--

<p>NOMBRE DE LA UNIDAD TEMÁTICA/BLOQUE TEMÁTICO:</p>	<p>Plantillas (Templates) y uso de la Biblioteca estándar de C++ con templates. Hilos y multihilos</p>	<p>TIEMPO ESTIMADO PARA DESARROLLAR LA UNIDAD TEMÁTICA:</p>	<p>10 horas (2 semanas)</p>
---	--	--	-----------------------------

COMPETENCIAS A DESARROLLAR	SABERES			EVIDENCIAS DE DESEMPEÑO	
	CONOCIMIENTOS	HABILIDADES	ACTITUDES	DIRECTA	POR PRODUCTO
1. Comprender el concepto de plantilla en C++ 2. Conocer y manipular el uso de plantillas en el código apropiado. 3. Revisar y realizar aplicaciones de funciones de la biblioteca estándar STL (Estándar Library with Templates). 4. Manejar y comprender el concepto de hilos y multihilos como herramienta para la programación en paralelo.	<ul style="list-style-type: none"> o Template (plantilla) o Declaración de clases usando templates. o Biblioteca Estándar de C (STL). 	<ul style="list-style-type: none"> - Comprender el concepto de plantilla en C++ - Conocer y manipular el uso de plantillas en el código apropiado. - Revisar y realizar aplicaciones de funciones de la biblioteca estándar STL (Estándar Library with Templates). - Manejar y comprender el concepto de hilos y multihilos como herramienta para la programación en paralelo 	<ul style="list-style-type: none"> • El fortalecimiento de hábitos correctos de estudio y análisis. • La valoración de la actividad creadora y la imaginación • La ética profesional al no falsificar información • La aceptación de los alcances y las limitaciones personales. • Proponer estrategias para la solución de problemas. • La valoración de la investigación interdisciplinaria y multidisciplinaria • La valoración de la actividad creadora y la imaginación 	1. Participación en clase 2. Ejercicios en pizarrón 3. Participación grupal en sesiones de ejercicios. 4. Evaluación mediante exámenes sorpresa de corta duración que cuenten como tarea. 5. Elaboración de prácticas de laboratorio	Tareas Exámenes Sorpresa Examen sumativo Cuaderno de ejercicios. Reporte de prácticas de laboratorio

NOMBRE DE LA UNIDAD TEMÁTICA/BLOQUE TEMÁTICO:	Programación de Interfaces Gráficas de Usuario	TIEMPO ESTIMADO PARA DESARROLLAR LA UNIDAD TEMÁTICA:	10 horas (2 semanas)
--	--	---	----------------------

COMPETENCIAS A DESARROLLAR	SABERES			EVIDENCIAS DE DESEMPEÑO	
	CONOCIMIENTOS	HABILIDADES	ACTITUDES	DIRECTA	POR PRODUCTO
1. Comprender el concepto de la programación de Interfaces gráficas para el usuario (GUI's) 2. Comprender el concepto de lanzamiento de eventos para distintos objetos gráficos.	<ul style="list-style-type: none"> o Interfaces gráficas para el usuario (GUI). o QT o Visual C++ 	<ul style="list-style-type: none"> - Comprender el concepto de la programación de Interfaces gráficas para el usuario (GUI's) - Comprender el concepto de lanzamiento de eventos para distintos objetos gráficos. 	<ul style="list-style-type: none"> • El fortalecimiento de hábitos correctos de estudio y análisis. • La valoración de la actividad creadora y la imaginación • La ética profesional al no falsificar información • La aceptación de los alcances 	1. Participación en clase 2. Ejercicios en pizarrón 3. Participación grupal en sesiones de ejercicios. 4. Evaluación	Tareas Exámenes Sorpresa Examen sumativo Cuaderno de ejercicios. Reporte de prácticas de laboratorio

<p>3. Conocer la forma de programación de GUI`s utilizando la herramienta QT de software libre.</p> <p>4. Conocer la forma de programación de GUI`s utilizando la herramienta Visual C++ de software libre.</p>		<ul style="list-style-type: none"> - Conocer la forma de programación de GUI`s utilizando la herramienta QT de software libre. - Conocer la forma de programación de GUI`s utilizando la herramienta Visual C++ de software libre. 	<ul style="list-style-type: none"> • y las limitaciones personales. • Proponer estrategias para la solución de problemas. • La valoración de la investigación interdisciplinaria y multidisciplinaria • La valoración de la actividad creadora y la imaginación 	<p>mediante exámenes sorpresa de corta duración que cuenten como tarea.</p> <p>5. Elaboración de prácticas de laboratorio</p>	
---	--	--	---	---	--

NOMBRE DE LA UNIDAD TEMÁTICA/BLOQUE TEMÁTICO:		Programación básica en LabView			TIEMPO ESTIMADO PARA DESARROLLAR LA UNIDAD TEMÁTICA:		10 horas (2 semanas)
COMPETENCIAS A DESARROLLAR	SABERES				EVIDENCIAS DE DESEMPEÑO		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES		DIRECTA		POR PRODUCTO
<p>1. Conocer la herramienta de desarrollo LabView.</p> <p>2. Desarrollar instrumentos virtuales con LabView.</p> <p>3. Realizar comunicaciones con dispositivos externos usando tarjetas de adquisición de datos con Labview.</p> <p>4. Realizar gráficas y manipulación de archivos usando LabView.</p>	<ul style="list-style-type: none"> o LabView. o Instrumento virtual. o Tarjetas de adquisición de datos. o Herramientas de manipulación de datos. 	<ul style="list-style-type: none"> - Conocer la herramienta de desarrollo LabView. - Desarrollar instrumentos virtuales con LabView. - Realizar comunicaciones con dispositivos externos usando tarjetas de adquisición de datos con Labview. - Realizar gráficas y manipulación de archivos usando LabView. 	<ul style="list-style-type: none"> • El fortalecimiento de hábitos correctos de estudio y análisis. • La valoración de la actividad creadora y la imaginación • La ética profesional al no falsificar información • La aceptación de los alcances y las limitaciones personales. • Proponer estrategias para la solución de problemas. • La valoración de la investigación interdisciplinaria y multidisciplinaria • La valoración de la actividad creadora y la imaginación 		<ol style="list-style-type: none"> 1. Participación en clase 2. Ejercicios en pizarrón 3. Participación grupal en sesiones de ejercicios. 4. Evaluación mediante exámenes sorpresa de corta duración que cuenten como tarea. 5. Elaboración de prácticas de laboratorio 		<p>Tareas</p> <p>Exámenes Sorpresa</p> <p>Examen sumativo</p> <p>Cuaderno de ejercicios.</p> <p>Reporte de prácticas de laboratorio</p>

ACTIVIDADES DE APRENDIZAJE (Sugeridas)

Elaboración de un cuaderno individual foliado para tareas.
Exposición del tema
Estudio en grupo para las sesiones de resoluciones de problemas.

RECURSOS Y MATERIALES DIDÁCTICOS (Sugeridos)

Recursos didácticos:
Pizarrón, computadora, cañón, bibliografía, red
Materiales didácticos:
Cuaderno de problemas.

SISTEMA DE EVALUACIÓN

EVALUACIÓN:

Será continua y permanente y se llevará a cabo en 2 momentos:

Formativa: Participación en clase, tareas, participación grupal en sesiones de solución de problemas.

Sumaria: exámenes escritos, entrega de cuaderno de tareas, autoevaluación, co-evaluación.

El ejercicio de autoevaluación y coevaluación tendrá el 5% de la ponderación individual, debido a que su finalidad es para retroalimentar el proceso formativo y ético del alumno.

PONDERACIÓN (SUGERIDA):

Revisión de cuaderno de problemas	30%
Participación individual	10%
Resultados de exámenes escritos	55%
Autoevaluación y coevaluación	5%

La calificación mínima aprobatoria será 7.0

FUENTES DE INFORMACIÓN

BIBLIOGRAFÍA BÁSICA:

1. Essential C++ Stanley B. Lippman Addison-Wesley Professional.
2. The C++ Standard Library: A Tutorial and Reference, Nicolai M. Josuttis. Addison-Wesley Professional.
3. Hands-On Introduction to LabVIEW for Scientists and Engineers, John Essick. Oxford University Press

BIBLIOGRAFÍA COMPLEMENTARIA:

1. The C++ Programming Language, Bjarne Stroustrup Addison-Wesley Professional; 3 ed.
2. Accelerated C++: Practical Programming by Example, Andrew Koenig, Barbara E. Moo. Addison-Wesley Professional.
3. Threads Primer: A Guide to Multithreaded Programming Bil Lewis, Daniel J. Berg Prentice Hall PTR.

OTRAS FUENTES DE INFORMACIÓN:

- Cientos de páginas web que proponen técnicas de aprendizaje de C++ , Visual C++, QT, Labview.