



PROGRAMA DE CURSO – Primer Semestre 2019

Curso:	Informática 2
Carrera:	Ingeniería en Sistemas
Catedrático:	Ernesto Rodríguez

No. de
períodos a la
semana:

Horario:

Área:

Requisito:

Requisito para:

4	Lunes, Martes 6:50 – 8:20 / Jueves 8:40-10:00	Troncal Profesional	Informática 1	
---	--	---------------------	---------------	--

Objetivos:

- Aprender a desarrollar software de calidad profesional utilizando los métodos y herramientas predominantes en la industria de software.
- Desarrollar y practicar las destrezas de análisis y razonamiento necesarias para convertir una idea o problema en una solución de software.
- Aprender principios importantes de la programación como: Única responsabilidad, programación con interfaces, abstracción, trabajo en equipo, desacoplamiento, etc.
- Familiarizar al estudiante con los mecanismos utilizados por las computadoras digitales/Von Neuman para representar, ordenar y procesar datos

Fecha	Objetivos comunes	Contenido	Actividad	Evaluación
Semana 1. 14 - 18 de Enero	Introducir al estudiante al lenguaje de programación C++ y familiarizar al estudiante con los conceptos y filosofía del lenguaje	Capítulos 2.1 y 2.2.1	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo</i>	Hoja de trabajo #1
Semana 2 21 - 25 de Enero	Familiarizar al estudiante con los tipos primitivos de datos y su representación en memoria. Familiarizar al estudiante con el concepto de una	Capítulo 2.2.2 y 2.2.3 <ul style="list-style-type: none">• Datos primitivos• variables• aritmética	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i>	Hoja de trabajo #2 Examen Corto #1

	variable, puntero y referencia..			
Semana 3 28 – 1 de Febrero	Familiarizar al estudiante con entrada y salida básica, estructuras de control if/switch, condicionales y ciclos while.	Capítulo 2.2.4 <ul style="list-style-type: none"> • Condicionales • Control de flujo • Entrada/Salida • Ciclos 	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i>	Hoja de trabajo #3 Examen corto #2
Semana 4 4 – 8 de Febrero	PRIMERA EVALUACIÓN PARCIAL Nota: en la semana de exámenes parciales se impartirán las clases normalmente. Último día para presentar nota de Primer Parcial: 15 de Febrero			
Semana 4 4 – 8 de Febrero	Familiarizar al estudiante con entrada y salida básica, estructuras de control if/switch, condicionales y ciclos while.	Capítulo 2.2.4 <ul style="list-style-type: none"> • Condicionales • Control de flujo • Entrada/Salida • Ciclos 	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i>	Hoja de trabajo #3 Examen corto #2
Semana 5 11 – 15 de Febrero	Introducir al estudiante al concepto de arreglos y repasar el concepto de punteros, referencias y direcciones de memoria aplicadas a arreglos.	Capítulo 2.2.5 <ul style="list-style-type: none"> • Arreglos • Iteración • Punteros • Memoria Capítulo 7	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i>	Hoja de trabajo #4 Examen corto #3
Semana 6 18 -22 de Febrero	Repasar el concepto de un “tipo” y mostrar al estudiante cómo puede definir sus propios tipos mediante estructuras y enumeraciones. Introducir el concepto de modularidad y encapsulación para mejor organización de código.	Capítulos 2.3.1-2.3.3 <ul style="list-style-type: none"> • Tipos • Estructuras • Tipos enumerados Capítulo 8.1, 8.2 y 8.3	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i>	Hoja de trabajo #5 Examen corto #4
Semana 7 25 de Febrero al 1 de Marzo	Familiarizarse con el concepto de clases y objetos y aprender a utilizar correctamente	Capítulos 2.3.1-2.3.3 y 2.4 - 2.4.2 <ul style="list-style-type: none"> • Clases • Modularidad 	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i>	Hoja de trabajo #6 Examen corto #5

	estos mecanismos de abstracción. Introducir el concepto de modularidad y encapsulación para mejor organización de código.			
Semana 8 4 – 8 de Marzo	SEGUNDA EVALUACIÓN PARCIAL Nota: en la semana de exámenes parciales se impartirán las clases normalmente. Último día para presentar nota de Segundo Parcial: 15 de Marzo			
Semana 8 4 de Febrero al 8 de Marzo	Familiarizarse con el concepto de clases y objetos y aprender a utilizar correctamente estos mecanismos de abstracción. Introducir el concepto de modularidad y encapsulación para mejor organización de código.	Capítulos 2.3.1-2.3.3 y 2.4 - 2.4.2 <ul style="list-style-type: none"> • Clases • Modularidad Capítulo 16	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i>	Hoja de trabajo #6 Examen corto #5
Semana 9 11 – 15 de Marzo	Introducir al estudiante a los mecanismos de abstracción que ofrece C++ y guiarlo en la utilización correcta de estos mecanismos.	Capítulos 3- 3.2.1 <ul style="list-style-type: none"> • Clases • Contenedoras • Manejo de memoria 	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i> <i>Develación del proyecto final</i>	Hoja de trabajo #7 Examen corto #6
Semana 10 18 – 22 de Marzo	Que el estudiante comprenda el concepto de tipos abstractos y los pueda utilizar para reducir la repetición de código.	Capítulo 3.2.2 <ul style="list-style-type: none"> • Tipos abstractos 	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i>	Hoja de trabajo #8 Examen corto #7
Semana 11 25 – 29 de Marzo	Familiarizar al estudiante con la herencia para que aprenda a crear clases con código re-utilizable. Discutir la múltiple herencia vs herencia única.	Capítulo 3.2.4 <ul style="list-style-type: none"> • Herencia • Métodos virtuales • Overriding Capítulo 21	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i>	Hoja de trabajo #9 Examen corto #8
Semana 12 1 – 5 de Abril	TERCERA EVALUACIÓN PARCIAL Nota: en la semana de exámenes parciales se impartirán las clases normalmente.			



	Último día para presentar nota de Tercer Parcial: 12 de Abril			
Semana 12 1 – 5 de Abril	Familiarizar al estudiante con las semánticas de copiado y movimiento para que el pueda entender como una computadora maneja, reserva y traslada la memoria.	Capítulo 3.3 <ul style="list-style-type: none"> • Copiado • Movimiento • Overriding de métodos 	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i>	Hoja de trabajo #9 Examen corto #8
Semana 13 8 – 12 de Abril	Familiarizar al estudiante con el concepto de plantillas las cuales le permitirán aplicar los conceptos de tipos genéricos aprendidos en Informática 1 en el lenguaje C++	Capítulo 3.4.1 <ul style="list-style-type: none"> • Plantillas • Tipos paramétricos • clases genéricas 	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i>	Hoja de trabajo #10 Examen corto #9
14– 21 de Abril	Semana Santa			
Semana 14 22 – 26 de Abril	Familiarizar al estudiante con otras aplicaciones de plantillas, funciones y métodos genéricos.	Capítulo 3.4.2 <ul style="list-style-type: none"> • Plantillas • Funciones genéricas Capítulo 23.1 - 23.4	<i>Catedrático: Clase magistral</i> <i>Alumno: hoja de trabajo y examen corto</i>	Hoja de trabajo #11 Examen corto #10
Semana 15 29 de Abril – 3 de Mayo 1 de Mayo asueto	Que el estudiante aprenda a expresar funciones como si fuesen objetos, de manera que pueda entender la relación entre datos y código	Capítulo 3.4.3 <ul style="list-style-type: none"> • Objetos como función 	<i>Catedrático: Clase magistral</i> <i>Alumno: Hoja de trabajo y examen corto</i>	Hoja de trabajo #12 Examen corto #11
Semana 16 6 – 10 de Mayo	EXÁMENES FINALES 17 de mayo. Último día de ingreso de notas FINALES al Blackboard.			

Evaluación:

Zona	60 puntos
Examen Final	40 puntos
TOTAL	100 PUNTOS

- Para tener derecho a Examen Final se requiere zona mínima de 30 puntos y 80% de asistencia
- Nota mínima para aprobar el curso 61 puntos
- Examen de Recuperación es sobre 40 puntos (conserva la zona)

Artículo	Detalles	Valor unitario	Total
Hoja de trabajo semanal	El estudiante trabajara en una hoja de trabajo semanal. Esta hoja ayudará al estudiante a repasar el contenido aprendido en clase. Durante el semestre habrán 12 hojas de trabajo, sin embargo solo las 10 mejores notas se tomarán en cuenta.	1%	10%
Examen corto	Antes de iniciar un tema nuevo cada semana, se realizará un examen corto de forma individual . Durante el semestre habrán 11 exámenes cortos de los cuales se tomarán las 9 mejores notas.	1%	9%
Participación en clase	Durante el semestre, se llevarán a cabo actividades dentro del salón de clase las cuales serán calificadas y entregadas de inmediato.	TBA	10%
Examen parcial	El contenido aprendido en clase se evaluará mediante tres exámenes parciales.	7%	21%
Proyecto final -- Parte I	Entrega intermedia del proyecto final	10%	10%
Examen final	Todo el material estudiado en el curso será evaluado mediante un examen final. Dicho examen tendrá una parte teórica que se	20% Teórico	



	elaborará individualmente y una parte práctica que será la entrega de un proyecto final. Este proyecto se puede realizar de forma individual o en parejas pero la nota es individual por alumno.	20% Práctico	40%
--	---	--------------	-----

Bibliografía:

- B. Stroustrup, The C++ Programming Language, 4ta Edición, Addison-Wesley (**Texto primario**)
- J. Smiley, Learn to Program with C++, 1ra Edición, McGraw Hill
- <http://www.cplusplus.com/>, Referencia del lenguaje C++ y sus bibliotecas