



PROGRAMA DEL CURSO

NOMBRE DEL CURSO: Introducción a la Programación y Computación 1

CODIGO:	0770	CREDITOS:	4
ESCUELA:	Ciencias y Sistemas	AREA A LA QUE PERTENECE:	Desarrollo de Software
PRE REQUISITO:	33 créditos y 0103 Matemática Básica 2	POST REQUISITO:	0771 Introducción a la Programación y Computación 2 0796 Lenguajes Formales y de Programación.
CATEGORIA:	Obligatorio	VIGENCIA:	Segundo Semestre 2019
CATEDRATICO (A):	Ver anexo	AUXILIAR:	Staff
EDIFICIO:	T-3 y T-7	SECCION:	A, B, C, D y E
SALÓN DEL CURSO:	Ver anexo	SALON DEL LABORATORIO:	Pendiente
HORAS POR SEMANA DEL CURSO:	4	HORAS POR SEMANA DEL LABORATORIO:	2
DÍAS QUE SE IMPARTE EL CURSO:	Martes y Jueves	DIAS QUE SE IMPARTE EL LABORATORIO:	Jueves
HORARIO DEL CURSO:	7:10 – 8:50	HORARIO DEL LABORATORIO:	9:10 – 10:50

DESCRIPCIÓN DEL CURSO:

El curso es el acercamiento inicial del estudiante de la carrera de sistemas, a la programación mediante el uso de disciplinas y metodologías especializadas. El curso se fundamenta en el concepto de algoritmo para la resolución de problemas de programación, enfatizando el uso del paradigma orientado a objetos. Se introducen conceptos básicos de UML como guía para el diseño de sistemas orientados a objetos. Se acerca al estudiante al conocimiento de los principales algoritmos de búsquedas y ordenamientos. Se cubre una parte importante de las estructuras de datos, los tipos de datos abstractos. Asimismo, el estudiante conocerá el lenguaje Java como el lenguaje oficial de programación del curso.

OBJETIVOS:

General

- Lograr que el estudiante adquiera la habilidad de programar independientemente de la plataforma y los conocimientos básicos de la programación utilizando el paradigma orientado a objetos.

Específico

1. Integrar al estudiante a la tecnología de la computación.
2. Conocer las diferentes metodologías de programación.
3. Adquirir la habilidad de hacer algoritmos.
4. Analizar los problemas con metodología orientada a objetos.
5. Aprender a elaborar diseños de clases preliminares en UML.
6. Organizar soluciones utilizando un lenguaje de programación.
7. Conocer el lenguaje Java como el primer lenguaje de programación para computadoras para una arquitectura stand-alone.

METODOLOGÍA:

- Clases diarias.
- Elaboración de investigaciones y tareas.
- Práctica de exámenes cortos y parciales.
- Laboratorio taller.

Elaboración de proyectos de programación.

EVALUACION DEL RENDIMIENTO ACADEMICO:

Clase teórica (70 puntos)		Clase práctica (30 puntos)	
Descripción	Pts.	Descripción	Pts.
Tareas, Cortos y Asistencia	5	Tareas	5
Primer parcial	12	Hojas de Trabajo	10
Segundo parcial	13	Prácticas	20
Tercer parcial	15	Proyectos	50
Laboratorio	30	Exámenes cortos	5
Zona total	75	Zona total	90
Examen Final	25	Examen Final	10
Total	100	Total	100

El curso se gana con 61 pts. de 100. Y el laboratorio de gana con 61 pts. de 100.

CONTENIDO:

<ol style="list-style-type: none"> 1. Introducción <ol style="list-style-type: none"> 1.1. Conceptos computacionales <ol style="list-style-type: none"> 1.1.1. Computadora 1.1.2. Hardware 1.1.3. Firmware 1.1.4. Software 1.2. Organización <ol style="list-style-type: none"> 1.2.1. CPU 1.2.2. Memoria principal 1.2.3. Memoria secundaria 1.2.4. Dispositivos E/S 1.2.5. Periféricos 1.3. Lenguajes de programación <ol style="list-style-type: none"> 1.3.1. Lenguaje de máquina 1.3.2. Lenguajes de bajo nivel 1.3.3. Lenguajes de alto nivel 1.4. Resolución de problemas computacionales <ol style="list-style-type: none"> 1.4.1. Análisis del problema 1.4.2. Diseño del algoritmo 1.4.3. Codificación 1.4.4. Compilación y ejecución 1.4.5. Verificación y depuración 1.4.6. Documentación 	<ol style="list-style-type: none"> 2. Programación modular y estructuras básicas <ol style="list-style-type: none"> 2.1. Secuencial y procedural: metodología Top-Down. 2.2. Variables: concepto, manipulación y asignación. 2.3. Tipos de datos (primitivos y construidos por el usuario) 2.4. Operadores aritméticos 2.5. Operadores relacionales y lógicos 2.6. Estructuras de control condicionales <ol style="list-style-type: none"> 2.6.1. Si – Sino (if – else)
---	---

- 2.6.2. En caso (switch / case)
- 2.7. Estructuras cíclicas (bucles, loops)
 - 2.7.1. Para (for)
 - 2.7.2. Mientras (while)
 - 2.7.3. Repetir - Hasta (Repeat – Until / do-while)
- 2.8. Las rutinas
 - 2.8.1. Procedimiento y función
 - 2.8.2. Entorno de las variables (alcance o ámbito)
 - 2.8.3. Los parámetros
 - 3.8.3.1 Por referencia
 - 3.8.3.2 Por valor
 - 2.8.4. El valor de retorno
- 2.9. Modularidad
 - 2.9.1. Segmentos por rutina
 - 2.9.2. Uso adecuado de prefijos
 - 2.9.3. Documentación interna
 - 2.9.4. Legibilidad y entendimiento
- 2.10. Recursividad

- 3. Metodología Orientada a Objetos
 - 3.1. Concepto de abstracción y clasificación
 - 3.2. Clases y objetos
 - 3.3. Mensajes y métodos
 - 3.4. El principio el encapsulamiento
 - 3.5. Los miembros de una clase
 - 3.5.1. Atributos
 - 3.5.2. Métodos (operaciones)
 - 3.5.3. Constructores y destructores
 - 3.6. Modificadores de visibilidad
 - 3.6.1. Privado
 - 3.6.2. Público
 - 3.6.3. Protegido
 - 3.7. Relaciones entre clases y objetos
 - 3.7.1. Asociación
 - 3.7.2. Agregación y composición
 - 3.7.3. Herencia (simple y múltiple)
 - 3.8. Polimorfismo
 - 3.8.1. Sobrecarga de métodos
 - 3.8.2. Virtualización
 - 3.9. Construcciones abstractas
 - 3.9.1. Clase abstracta
 - 3.9.2. Interface
 - 3.10. Conceptos avanzados
 - 3.10.1. Miembros estáticos (static) y miembros de instancia
 - 3.10.2. Referencia “this”
 - 3.10.3. Clases paramétricas (plantilla de clases).
 - 3.11. Principios básicos de UML (diagrama de clases)
 - 3.11.1. Definición de clases y sus relaciones
 - 3.11.2. Ámbito de las propiedades, Métodos
 - 3.11.3. Diseño de programas
 - 3.11.4. Asociaciones y restricciones, clases de asociaciones, Multiplicidad, Dependencia
 - 3.11.5. Relaciones múltiples (asociativas) y reflexivas

- 4. Estructuras algorítmicas
 - 4.1. Arreglos vectoriales de datos
 - 4.1.1. Conceptos: elementos, longitud, indexación, representación en memoria.
 - 4.1.2. Arreglos bidimensionales (matrices): representación en memoria.

	<ul style="list-style-type: none">4.1.3. Arreglos n-dimensionales (multidimensionales).4.2. Las cadenas de caracteres<ul style="list-style-type: none">4.2.1. Concepto: diferencia con arreglos de caracteres.4.2.2. Cadenas estáticas (ej: String) y dinámicas (ej: StringBuffer).4.2.3. Operaciones y métodos.4.3. Búsqueda de datos en arreglos<ul style="list-style-type: none">4.3.1. Secuencial4.3.2. Binaria4.4. Ordenamiento de datos en arreglos<ul style="list-style-type: none">4.4.1. Burbuja4.4.2. Por inserción4.4.3. Por selección4.4.4. Quick Sort4.5. La pila (Stack)<ul style="list-style-type: none">4.5.1. Política de acceso a datos (LIFO) y operaciones.4.6. La cola (Queue)<ul style="list-style-type: none">4.6.1. Política de acceso a datos (FIFO) y operaciones.4.6.2. Representaciones: simple y circular.4.7. El uso de Heap<ul style="list-style-type: none">4.7.1. Asociación a la pila4.7.2. Tomar y devolver al heap4.7.3. Usos con las pilas y las colas
	<ul style="list-style-type: none">5. Colecciones de Datos y tipos de Datos Abstractos<ul style="list-style-type: none">5.1. Los índices y el apuntador simple5.2. El apuntador subíndice5.3. Almacenamiento5.4. Ordenamiento5.5. Los registros5.6. Concepto y definición por campos5.7. Tipos de apuntadores (estáticos y dinámicos)5.8. Listas simples5.9. Listas doblemente encadenadas5.10. Pilas usando listas5.11. Colas usando listas5.12. Listas ortogonales5.13. Listas n-encadenadas
	<ul style="list-style-type: none">6. Flujos de bytes y manipulación de archivos<ul style="list-style-type: none">6.1. Concepto: modelo productor-consumidor y flujo (stream).6.2. Tipos de archivos6.3. Archivos de texto6.4. Operaciones básicas<ul style="list-style-type: none">6.4.1. Abrir y cerrar6.4.2. Lectura, escritura y posicionamiento6.4.3. Localización del final del archivo
	<ul style="list-style-type: none">7. Cloud Computing (Investigación requisito para examen final)<ul style="list-style-type: none">7.1. Antecedentes e Historia7.2. Definición7.3. Funcionamiento7.4. Tipos de nube7.5. Beneficios de la nube
7.6.	<ul style="list-style-type: none">Servicios de la nube<ul style="list-style-type: none">7.6.1. Modelo SaaS7.6.2. Modelo PaaS7.6.3. Modelo IaaS7.7. Aplicaciones prácticas7.8. Empresas relacionadas

CLÁUSULAS RESTRICTIVAS:

El perfil del estudiante de la facultad de Ingeniería de la Universidad de San Carlos de Guatemala exige una alta calidad en la excelencia académica y ética profesional. Se establecen en este curso los siguientes lineamientos que regulan el comportamiento del estudiante:

- Copias en exámenes, cortos, proyectos, tareas e investigaciones tienen cero de nota.
- Exámenes parciales y examen final NO tienen reposición.
- No hay prorrogas.
- No hay reposición de proyectos.
- Cualquier proyecto, tarea o investigación que se entregue después de la fecha calendarizada tiene 30 puntos menos, cada día de atraso.
- Los exámenes resueltos a lápiz no tienen derecho a revisión.
- Es obligatorio ganar el laboratorio para tener derecho a evaluación total del curso.
- Para poder optar a la revisión de la zona final es obligatorio haber asistido a los exámenes parciales y al examen final.

BIBLIOGRAFÍA:

- JOYANES, L. y ZAHONERO, I. “**Programación en Java 2 (algoritmos, estructura de datos y programación orientada a objetos)**”. España, McGraw-Hill / Interamericana de España, S. A. 2002, PP 725
- JOYANES, L. “**Programación en Turbo Pascal Versiones 5.5, 6.0, y 7.0**”, (2da Edición), México, McGraw-Hill / Interamericana de España, S. A. 1995, PP. 914
- Deitel & Deitel. “**Cómo Programar en Java**” (7ma Edición), México, Prentice Hall 2008, PP. 1280
- McLaughlin, B.; Pollice, G. y West, D. “**Head First Object-Oriented Analysis & Design**”, EUA, O’Reilly Media 2006, PP. 636
- Freeman, E.; Robson, E.; Bates, B. y Sierra, K. “**Head First Design Patterns**”, EUA, O’Reilly Media 2004, PP. 694
- Manuales de Referencia de Java, <<http://www.sun.com/java>>.
- Cualquier otro material (escrito o digital) entregado en clase.

LISTA DE CATEDRÁTICOS

CURSO	SEC	EDIFICIO	SALON	CATEDRATICOS
Introducción a la Programación y Computación 1	A	T-3	215	Marlon Francisco Orellana Lopez
Introducción a la Programación y Computación 1	B	T-3	014	Byron Rodolfo Zepeda Arévalo
Introducción a la Programación y Computación 1	C	T-7	202	Moises Eduardo Velasquez Oliva
Introducción a la Programación y Computación 1	D	T-3	310	Herman Igor Veliz Linares
Introducción a la Programación y Computación 1	E	T-3	105	Neftali De Jesus Calderon Mendez

Julio									
Lunes		Martes		Miércoles		Jueves		Viernes	
	1		2		3		4		5
	8		9		10		11		12
	15	<ul style="list-style-type: none"> • Generación de listados de estudiantes y balanceo de secciones • Conceptos computacionales • Organización • Lenguajes de programación 	16		17	<ul style="list-style-type: none"> • Resolución de problemas computacionales 	18		19
	22	<ul style="list-style-type: none"> • Resolución de problemas computacionales 	23		24	<ul style="list-style-type: none"> • Secuencial y procedural: metodología Top-Down. • Variables: concepto, manipulación y asignación. • Tipos de datos (primitivos y construidos por el usuario) • Operadores aritméticos 	25		26
	29	<ul style="list-style-type: none"> • Operadores relacionales y lógicos • Estructuras de control condicionales • Estructuras cíclicas (bucles, loops) 	30		31				

Agosto									
Lunes		Martes		Miércoles		Jueves		Viernes	
							1		2
						<ul style="list-style-type: none"> • Las rutinas • Modularidad • Recursividad 			
	5	<ul style="list-style-type: none"> • Recursividad 	6		7	<ul style="list-style-type: none"> • Actividad 	8		9
	12	<ul style="list-style-type: none"> • Concepto de abstracción y clasificación • Clases y objetos • Mensajes y métodos • El principio el encapsulamiento 	13		14	<ul style="list-style-type: none"> • Feriado 	15		16

19	20	21	22	23
	<ul style="list-style-type: none"> • Los miembros de una clase • Modificadores de visibilidad • Relaciones entre clases y objetos • Polimorfismo 		<ul style="list-style-type: none"> • Construcciones abstractas • Conceptos avanzados • Principios básicos de UML (diagrama de clases) 	
26	27	28	29	30
	<ul style="list-style-type: none"> • Diagrama de clases 		<ul style="list-style-type: none"> • Diagrama de clases 	

Septiembre				
Lunes	Martes	Miércoles	Jueves	Viernes
2	3	4	5	6
	<ul style="list-style-type: none"> • Arreglos vectoriales de datos • Las cadenas de caracteres • Búsqueda de datos en arreglos 		<ul style="list-style-type: none"> • Búsqueda de datos en arreglos 	
9	10	11	12	13
	<ul style="list-style-type: none"> • Ordenamiento de datos en arreglos 		<ul style="list-style-type: none"> • Ordenamiento de datos en arreglos 	
16	17	18	19	20
	<ul style="list-style-type: none"> • La pila (Stack) • La cola (Queue) • El uso de Heap 		<ul style="list-style-type: none"> • Actividad 	
23	24	25	26	27
Semana de Congressos	Semana de Congressos	Semana de Congressos	Semana de Congressos	Semana de Congressos
30				

Octubre				
Lunes	Martes	Miércoles	Jueves	Viernes
	1	2	3	4
	<ul style="list-style-type: none"> • Aplicación de pilas y colas 		<ul style="list-style-type: none"> • Aplicación de pilas y colas 	
7	8	9	10	11
	<ul style="list-style-type: none"> • Colecciones de Datos y tipos de Datos Abstractos 		<ul style="list-style-type: none"> • Colecciones de Datos y tipos de Datos Abstractos 	
14	15	16	17	18
	<ul style="list-style-type: none"> • Colecciones de Datos y tipos de Datos Abstractos 		<ul style="list-style-type: none"> • Aplicación de memoria dinámica 	
21	22	23	24	25
	<ul style="list-style-type: none"> • Aplicación de memoria dinámica 		<ul style="list-style-type: none"> • Actividad 	
28	29	30	31	
	<ul style="list-style-type: none"> • Flujos de bytes y manipulación de archivos 		<ul style="list-style-type: none"> • Flujos de bytes y manipulación de archivos 	

Noviembre									
Lunes		Martes		Miércoles		Jueves		Viernes	
									1
	4		5		6		7		8
	11		12		13		14		15
	18		19		20	• Examen Final (Sujeta a calendario)			22
	25		26		27		28		29