

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
 FACULTAD DE INGENIERIA  
 ESCUELA DE CIENCIAS y SISTEMAS  
**PROGRAMA DEL CURSO**



<b>I. INFORMACIÓN GENERAL</b>			
<b>NOMBRE DEL CURSO:</b> Introducción a la Programación y Computación 1			
<b>CODIGO:</b>	0770	<b>CREDITOS:</b>	4
<b>ESCUELA:</b>	Ciencias y Sistemas	<b>AREA A LA QUE PERTENECE:</b>	Desarrollo de Software
<b>PRE REQUISITO:</b>	34 créditos y 0103 Matemática Básica 2	<b>POST REQUISITO:</b>	* 0771 Introducción a la Programación y Computación 2 * 0796 Lenguajes Formales y de Programación. * 0962 Matemática para Computación 2
<b>CATEGORIA:</b>	Obligatorio	<b>SEMESTRE:</b>	2do. Semestre 2018
<b>CATEDRÁTICO (S):</b>	Ing. Moisés Velásquez Ing. Neftalí Calderon Ing. William Escobar Ing. Herman Veliz Ing. Marlon Orellana	<b>AUXILIAR(ES):</b>	José Portillo Robin Salvatierra
		<b>SALÓN DEL CURSO:</b>	Según horario Oficial
<b>HORARIO DEL CURSO</b>	: 7:00 AM – 8:50 AM	<b>Días:</b>	Martes y Jueves

**II. 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.

**III. OBJETIVOS:**

**General**

- Lograr que el estudiante adquiriera la habilidad de programar 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. Organizar soluciones utilizando un lenguaje de programación.
4. Adquirir la habilidad de hacer algoritmos.
5. Aprender a elaborar diseños de clases preliminares en UML.
6. Analizar los problemas con metodología orientada a objetos.
7. Conocer el lenguaje Java como el primer lenguaje de programación para computadoras.

#### IV. METODOLOGÍA:

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

#### V. EVALUACION DEL RENDIMIENTO ACADÉMICO:

Clase teórica (70 puntos)		Clase práctica (30 puntos)	
Descripción	Pts.	Descripción	Pts.
Tareas, Cortos y Asistencia	09	Tareas	06
Primer parcial	08	Evaluaciones	06
Segundo parcial	14	Practica Presencial	09
Tercer parcial	14	Prácticas	20
Laboratorio	30	Proyectos	50
	-----		-----
Zona total	75	Zona total	91
Examen Final	25	Evaluación Final	09
	-----		-----
Total	100	Total	100

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

#### VI. CONTENIDO

1. Introducción
  - 1.1 Conceptos computacionales
    - 1.1.1 Computadora
    - 1.1.2 Hardware
    - 1.1.3 Firmware
    - 1.1.4 Software
  - 1.2 Organización
    - 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
    - 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
    - 1.4.1 Toma de requerimientos
      - 1.4.1.1 Plan de requerimientos
    - 1.4.2 Análisis del problema
    - 1.4.3 Diseño del algoritmo
    - 1.4.4 Codificación
    - 1.4.5 Compilación y ejecución
    - 1.4.6 Verificación y depuración
    - 1.4.7 Documentación
      - 1.4.7.1 Plan de proyectos de software
      - 1.4.7.2 Estrategias de Marketing
2. Metodología orientada a objetos
  - 2.1 Concepto de abstracción y clasificación
  - 2.2 Clases y objetos
  - 2.3 Mensajes y métodos

- 2.4 El principio el encapsulamiento
- 2.5 Los miembros de una clase
  - 2.5.1 Atributos
  - 2.5.2 Métodos (operaciones)
  - 2.5.3 Constructores y destructores
- 2.6 Modificadores de visibilidad
  - 2.6.1 Privado
  - 2.6.2 Público
  - 2.6.3 Protegido
- 2.7 Relaciones entre clases y objetos
  - 2.7.1 Asociación
  - 2.7.2 Agregación y composición
  - 2.7.3 Herencia (simple y múltiple)
- 2.8 Polimorfismo
  - 2.8.1 Sobrecarga de métodos
  - 2.8.2 Virtualización
- 2.9 Construcciones abstractas
  - 2.9.1 Clase abstracta
  - 2.9.2 Interfase
- 2.10 Conceptos avanzados
  - 2.10.1 Miembros estáticos (static) y miembros de instancia
  - 2.10.2 Referencia "this"
  - 2.10.3 Clases paramétricas (plantilla de clases).
- 2.11 Principios básicos de UML (diagrama de clases)
  - 2.11.1 Definición de clases y sus relaciones
  - 2.11.2 Ámbito de las propiedades, Métodos
  - 2.11.3 Diseño de programas
  - 2.11.4 Asociaciones y restricciones, clases de asociaciones, Multiplicidad, Dependencia
  - 2.11.5 Relaciones múltiples (asociativas) y reflexivas
- 3. Programación modular y estructuras básicas
  - 3.1 Secuencial y procedural: metodología Top-Down.
  - 3.2 Variables: concepto, manipulación y asignación.
  - 3.3 Tipos de datos (primitivos y construidos por el usuario)
  - 3.4 Operadores aritméticos
  - 3.5 Operadores relacionales y lógicos
  - 3.6 Estructuras de control condicionales
    - 3.6.1 Si – Sino (if – else)
    - 3.6.2 En caso (switch / case)
  - 3.7 Estructuras cíclicas (bucles, loops)
    - 3.7.1 Para (for)
    - 3.7.2 Mientras (while)
    - 3.7.3 Repetir - Hasta (Repeat – Until / do-while)
  - 3.8 Las rutinas
    - 3.8.1 Procedimiento y función
    - 3.8.2 Entorno de las variables (alcance o ámbito)
    - 3.8.3 Los parámetros
      - 3.8.3.1 Por variables
      - 3.8.3.2 Por valor
    - 3.8.4 El valor de retorno
  - 3.9 Modularidad
    - 3.9.1 Segmentos por rutina
    - 3.9.2 Uso adecuado de prefijos
    - 3.9.3 Documentación interna
    - 3.9.4 Legibilidad y entendimiento
  - 3.10 Recursividad

4. Programación orientada a objetos – Laboratorio
  - 4.1 Lenguaje Java (clases, atributos, métodos)
  - 4.2 Constructor y destructor
  - 4.3 Tipos de atributos
  - 4.4 Operaciones (aritméticos, relacionales y lógicos)
  - 4.5 Estructuras de control condicionales (if – else, switch, ?:)
  - 4.6 Estructuras cíclicas (for, while, do-while)
  - 4.7 Tipos de accesos (public, private, protected)
  - 4.8 Manejo de variables.
  - 4.9 Métodos: funciones/procedimientos y recursividad.
  
5. Estructuras algorítmicas
  - 5.1 Arreglos vectoriales de datos
    - 5.1.1 Conceptos: elementos, longitud, indexación, representación en memoria.
    - 5.1.2 Arreglos bidimensionales (matrices): representación en memoria.
    - 5.1.3 Arreglos n-dimensionales (multidimensionales).
    - 5.1.4 Ejemplos, técnicas de acceso y recomendaciones.
  - 5.2 Las cadenas de caracteres
    - 5.2.1 Concepto: diferencia con arreglos de caracteres.
    - 5.2.2 Cadenas estáticas (ej: String) y dinámicas (ej: StringBuffer).
    - 5.2.3 Operaciones y métodos.
  - 5.3 Búsqueda de datos en arreglos
    - 5.3.1 Secuencial
    - 5.3.2 Binaria
  - 5.4 Ordenamiento de datos en arreglos
    - 5.4.1 Burbuja
    - 5.4.2 Por inserción
    - 5.4.3 Por selección
    - 5.4.4 Quick Sort
  - 5.5 La pila (Stack)
    - 5.5.1 Política de acceso a datos (LIFO) y operaciones.
  - 5.6 La cola (Queue)
    - 5.6.1 Política de acceso a datos (FIFO) y operaciones.
    - 5.6.2 Representaciones: simple y circular.
  - 5.7 El uso de Heap
    - 5.7.1 Asociación a la pila
    - 5.7.2 Tomar y devolver al heap
    - 5.7.3 Usos con las pilas y las colas
  
6. Colecciones de datos
  - 6.1 Los índices y el apuntador simple
    - 6.1.1 El apuntador subíndice
    - 6.1.2 Almacenamiento
    - 6.1.3 Ordenamiento
  - 6.2 Los registros
    - 6.2.1 Concepto y definición por campos
  
7. Flujos de bytes y manipulación de archivos
  - 7.1 Concepto: modelo productor-consumidor y flujo (stream).
  - 7.2 Tipos de flujos
  - 7.3 Tipos de archivos
    - 7.3.1 Archivos de texto
    - 7.3.2 Archivos binarios
  - 7.4 Operaciones básicas
    - 7.4.1 Abrir y cerrar
    - 7.4.2 Lectura, escritura y posicionamiento
    - 7.4.3 Localización del final del archivo

8. Los tipos de datos abstractos
  - 8.1 Tipos de apuntadores (estáticos y dinámicos)
  - 8.2 Listas simples
  - 8.3 Listas doblemente encadenadas
  - 8.4 Pilas usando listas
  - 8.5 Colas usando listas
  - 8.6 Listas ortogonales
  - 8.7 Listas n-encadenadas

#### VII. 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 sustentar cada uno de los exámenes parciales deberá entregarse completamente resuelta cada una de las tareas especiales pre-examen.
- Para poder optar a la revisión de la zona final es obligatorio haber asistido a dos exámenes parciales y al examen final.

#### VIII. 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.

#### Secciones de IPC 1

Sección A	Ing. Marlon Orellana	T7	201	07:10 - 08:50 Ma - Ju
Sección B	Ing. Byron Zepeda	T7	102	07:10 - 08:50 Ma - Ju
Sección C	Ing. Moisés Velásquez	T3	114	07:10 - 08:50 Ma - Ju
Sección D	Ing. Herman Veliz	T3	404	07:10 - 08:50 Ma - Ju
Sección E	Ing. Nefthalí Calderón	T7	101	07:10 - 08:50 Ma - Ju