



## PROGRAMA DEL CURSO

### I. Información General

Nombre	Introducción a la Programación y Computación 1
Código	0770
Créditos	4
Vigencia	Primer semestre 2015
Horario	2 períodos (martes y jueves) de 7:10 AM a 8:50 AM
Lugar	T7; 202
Sección	E
Prerrequisitos	34 créditos y 0103 Matemática Básica 2
Post requisitos	0771 Introducción a la Programación y Computación 2 0796 Lenguajes Formales y de Programación
Catedrático	Ing. Ing. Moisés Eduardo Velásquez Oliva
Auxiliares	Staff

### II. Descripción

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.

#### Específicos

- Integrar al estudiante a la tecnología de la computación.
- Conocer las diferentes metodologías de programación.
- Organizar soluciones utilizando un lenguaje de programación.
- Adquirir la habilidad de hacer algoritmos.
- Aprender a elaborar diseños de clases preliminares en UML.
- Analizar los problemas con metodología orientada a objetos.
- 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. Evaluación

Clase teórica (70 puntos)

Descripción	Pts
Tareas, Cortos y Asistencia	6
Primer parcial	13
Segundo parcial	13
Tercer parcial	13
Examen Final	25

Clase práctica (30 puntos) : Laboratorio

Descripción	Pts
Tareas, Cortos	2.4
Practicas	12
Proyecto	13.5
Examen Final	2.1

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 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
- 2 Programación modular y estructuras básicas
  - 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
    - 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
      - 2.8.3.1 Por variables
      - 2.8.3.2 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 del 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 Interfase
  - 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 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 tiene cero de nota.
- La reposición de cualquier parcial se hará tomando la misma nota del examen siguiente, siempre y cuando no tenga más de 4 inasistencias a clase y se justifique, debidamente comprobado, por escrito.
- El examen final NO tiene reposición.
- NO hay prórrogas.
- NO hay reposición de proyectos.
- Cualquier proyecto, tarea o investigación que se entregue después de la fecha calendarizada tiene 30 puntos menos por 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 tarea 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
- BUDD, Timothy. **“Introducción a la programación orientada a objetos”**, EUA, Addison-Wesley, Iberoamericana, S. A. 1994, PP. 409
- 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
- Manuales de Referencia de Java, <<http://www.sun.com/java>>.
- Cualquier otro material (escrito o digital) entregado en clase.