UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS

PROGRAMA DE LABORATORIO	
Nombre del curso:	Introducción a la programación y computación 1

Código:	0770	**Créditos**	4
Escuela:	Ciencias y Sistemas	**Área a la que pertenece:**	Desarrollo de Software
Prerequisito	33 créditos y 0103 Matemática Básica 2	**Post requisitos:**	0771 Introducción a la Programación y computación 2 0796 Lenguajes Formales y de programación
Categoría:	Obligatorio	**Vigencia:**	Primer Semestre 2026
Catedrático:	Ver anexo	**Auxiliar:**	Ver anexo
Edificio:	T1, T3, Virtual	**Secciones**	A, B, C, D, E, F, G
Salon del curso:	L-III-8: T1, 309, 310, 314, 316, 310: T3, Meet: Virtual	**Salon del laboratorio:**	Meet
Horas por sema del curso magistral:	4	**Horas por semana del laboratorio:**	2
Horario del curso magistral:	A, F: Lunes y Miércoles B,C,D,E: Martes y Jueves G: Sábado	**Día que se imparte el laboratorio:**	B, C, E, F, G: Jueves A, D: Viernes
Horarios del curso magistral:	07:10 - 08:50	**Horario del laboratorio:**	DEPENDE POR SECCIÓN

Descripción del curso
El curso busca ser el acercamiento inicial del estudiante de la carrera de sistemas, al mundo de Desarrollo de Software mediante el uso de métodos, técnicas y metodologías especializadas. Se fundamenta en el concepto de algoritmo para la resolución de problemas de programación utilizando computadoras, enfatizando el uso del paradigma de Programación Orientado a Objetos. Se acerca al estudiante al conocimiento de los principales algoritmos de búsquedas y ordenamientos. Asimismo, el estudiante conocerá el lenguaje Java como el lenguaje oficial de programación del curso.

Objetivos:

General

- Adquirir, por parte del estudiante, la destreza de analizar, diseñar y codificar software de alta calidad independientemente de la plataforma y lenguaje de programación fundamentado en los conocimientos básicos de la programación utilizando el Paradigma Orientado a Objetos.

Específicos:

1. Integrar al estudiante a la tecnología de la computación.
2. Conocer las diferentes metodologías de software.
3. Analizar los problemas bajo la perspectiva de Programación Orientada a Objetos.
4. Diseñar soluciones elegantes basadas en el entendimiento de proceso de Análisis
5. Organizar soluciones utilizando un lenguaje de programación oficial y complementario.

Metodología:

- Clases presenciales (días, ver inicio), usando el salón asignado con apoyo de la plataforma UEDI.
- Elaboración de investigaciones y tareas.
- Práctica de exámenes cortos y parciales.
- Laboratorio y talleres.
- Elaboración de proyectos de programación
- Elaboración de prácticas cortas de programación
- Cursos complementarios extra aula

Requisitos:

- El desarrollo de las actividades es de carácter individual. Todas las entregas serán evaluadas por copias entre secciones. Las copias parciales o totales tienen nota de 0 y reporte a la Escuela de Ciencias y Sistemas.
- El laboratorio se aprueba con 61 puntos.
- Las actividades por realizar en el laboratorio (tareas, prácticas, y proyectos) estarán coordinadas entre secciones.
- La forma de entrega de las actividades será vía UEDI, según la fecha y hora límite de entrega, indicada en el enunciado de cada actividad.
- Para la calificación de las actividades se tomará en cuenta la presentación, calidad, tolerancia.

Aislamiento de rendimiento académico			
Descripción	**Publicación**	**Entrega**	**Punteo**
Practica 1:	01/02/2024	14/02/2024	10
Practica 2:	07/03/2024	20/03/2024	15
Total de prácticas:			**25**
Proyecto 1:	15/02/2024	06/03/2024	20

Proyecto 2:	21/03/2024	25/04/2024	25
Total de proyectos:			**45**
Tarea 1:	25/01/2024	01/02/2024	2.5
Tarea 2:	08/02/2024	15/02/2024	2.5
Tarea 3:	29/02/2024	07/03/2024	2.5
Tarea 4:	04/04/2024	11/04/2024	2.5
Total tareas:			**10**
Corto 1	—	22/02/2024	5
Corto 2	—	04/04/2024	5
Total de Cortos:			**10**
Examen Final (02/05/2024)			**10**
Total: El laboratorio se gana con 61 pts. de 100. **Para ganar el laboratorio se debe de contar con un 80% de asistencia.**			**100**

Contenido

1. **Fundamentos de Programación y JAVA**
   - 1.1. Algoritmos
   - 1.2. ¿Qué es Java?
   - 1.3. Versiones y Ambiente de Java (JDK, JRE, JVM).
   - 1.4. Características de Java
   - 1.5. Comentarios de una línea y multilínea
   - 1.6. Variables
   - 1.7. Tipos Primitivos y No Primitivos
   - 1.8. Casteos Implicitos y Explicitos
   - 1.9. Operadores Aritméticos, Relacionales y Lógicos
   - 1.10. Prioridad entre operadores
   - 1.11. Input y output
   - 1.12. Estructuras de Control
     - 1.12.1. if, else if, else
     - 1.12.2. switch
   - 1.13. Ciclos
     - 1.13.1. for
     - 1.13.2. while
     - 1.13.3. do - while
   - 1.14. Arreglos y Listas Dinámicas
   - 1.15. Procedimientos y Funciones
   - 1.16. Recursividad
     - 1.16.1. Recursividad Simple
     - 1.16.2. Recursividad Indirecta
   - 1.17. Manejo de memoria
     - 1.17.1. Stack (Memoria Estática)
     - 1.17.2. Heap (Memoria Dinámica)
   - 1.18. Metodos de Ordenamiento:

- 1.18.1. Burbuja
- 1.18.2. Por inserción
- 1.18.3. Por Selección
- 1.18.4. Quick Sort

2. Versionamiento

- 2.1. Introducción a versionamiento
- 2.2. Herramientas de control de versiones
  - 2.2.1. Git
  - 2.2.2. Github, Gitlab
  - 2.2.3. Git Kraken
- 2.3. Tareas básicas
  - 2.3.1. Creación de repositorio
  - 2.3.2. Commit
    - 2.3.2.1. Working directory
    - 2.3.2.2. Staging area
    - 2.3.2.3. Repository
  - 2.3.3. Creación de ramas
- 2.4. Colaboración en repositorios remotos
  - 2.4.1. Clonación de repositorio
  - 2.4.2. Resolución de conflictos

3. Manejo de errores, debug y pruebas

- 3.1. Expresiones (Try-Catch, etc.)
- 3.2. Debugging
  - 3.2.1. Breakpoint
    - 3.2.1.1. Inline
    - 3.2.1.2. Function
      - 3.2.1.2.1. Entrar a función
      - 3.2.1.2.2. Salir de función
  - 3.2.2. Start
  - 3.2.3. Pause
  - 3.2.4. Continue
  - 3.2.5. Stop
- 3.3. Testing
  - 3.3.1. Unit test
  - 3.3.2. Functional test
  - 3.3.3. Integration test

4. Programación Orientada a Objetos (POO)

- 4.1. Concepto de abstracción y clasificación
- 4.2. Clases y objetos
- 4.3. Mensajes y métodos
- 4.4. El principio el encapsulamiento
- 4.5. Los miembros de una clase
  - 4.5.1. Atributos
  - 4.5.2. Métodos (operaciones)
  - 4.5.3. Constructores y Destructores
- 4.6. Modificadores de visibilidad
  - 4.6.1. Privado
  - 4.6.2. Público
  - 4.6.3. Protegido
- 4.7. Relaciones entre clases y objetos
  - 4.7.1. Asociación
  - 4.7.2. Agregación y composición
  - 4.7.3. Herencia
- 4.8. Polimorfismo
  - 4.8.1. Sobrecarga de métodos
  - 4.8.2. Virtualización

- 4.9. Construcciones abstractas
  - 4.9.1. Clase abstracta
  - 4.9.2. Interface
- 4.10. Conceptos avanzados
  - 4.10.1. Miembros estáticos (static) y miembros de instancia
  - 4.10.2. Referencia “this”
  - 4.10.3. Clases paramétricas (plantilla de clases)
- 4.11. Principios básicos de UML (diagrama de clases)
  - 4.11.1. Definición de clases y sus relaciones
  - 4.11.2. Ámbito de las propiedades, Métodos
  - 4.11.3. Diseño de programas
  - 4.11.4. Asociaciones y restricciones, clases de asociaciones, multiplicidad, dependencia.
  - 4.11.5. Relaciones múltiples (asociativas) y reflexivas

5. Interfaces Gráficas en JAVA

- 5.1. Librerías de interfaz gráfica AWT y SWING
- 5.2. Componentes de interfaz gráfica
- 5.3. Disparadores de Eventos

6. Concurrencia y Paralelismo

- 6.1. Procesos
- 6.2. Subprocesos
- 6.3. Hilos
  - 6.3.1. Método Start
  - 6.3.2. Detener hilo
  - 6.3.3. Esperá de finalización de un hilo
  - 6.3.4. Condición de carrera
- 6.4. Hilos en Java
- 6.5. Animación usando hilos

7. Manejo de Archivos

- 7.1. Archivos de texto plano
  - 7.1.1. Creación
  - 7.1.2. Lectura
  - 7.1.3. Escritura
  - 7.1.4. Eliminación
- 7.2. Serialización de objetos en archivos

8. Programación Web

- 8.1. Frontend
  - 8.1.1. HTML, CSS y Javascript
  - 8.1.2. Typescript
  - 8.1.3. Frontend Framework (React, Angular, etc.)
- 8.2. Backend
  - 8.2.1. Protocolo HTTP
  - 8.2.2. API REST
    - 8.2.2.1. Peticiones GET, POST, PUT, DELETE
  - 8.2.3. Node JS
  - 8.2.4. Backend Framework (Express JS)

9. Cloud Computing

- 9.1. Ventajas y Desventajas
- 9.2. Tipos de nube
  - 9.2.1. Pública
  - 9.2.2. Privada
  - 9.2.3. Hibrida
- 9.3. Modelos en la nube
  - 9.3.1. SaaS

9.3.2. PaaS
9.3.3. IaaS
9.4. Servicios en la nube
9.5. Proveedores de nube

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 prácticas y proyectos.
- No hay prórrogas.
- No hay reposición de proyectos.

Es obligatorio ganar el laboratorio para tener derecho a evaluación total del curso.

Puntos importantes a considerar:

- Para tener derecho a nota de laboratorio se debe cumplir con el 80% de asistencia a clase de laboratorio, a excepción de presentar una justificación y constancia.
- No se aceptarán entregas tarde sobre tareas, prácticas, exámenes cortos, exámenes finales y proyectos sin justificación. El tutor académico puede aplicar la penalización que considere apropiada.
- El medio de entrega oficial para las actividades es la plataforma UEDI de la facultad y formulario de Google proporcionado en el enunciado de cada actividad.
- Todo proyecto será verificado para validar la creación de este.
- Se realizará un seguimiento a las dudas planteadas en laboratorio sobre prácticas o proyectos.
- Copias obtendrán una nota de 0 y reportará a la Escuela de Ciencias y Sistemas.

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, P. 409
- 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
- Manuales de Referencia de Java, <http://www.sun.com/java>.
- Cualquier otro material (escrito o digital) entregado en clase.

Sección	Catedrático	Auxiliar
A	Manuel Haroldo Castillo Reyna	Rodrigo Alejandro Hernández de León
B	William Estuardo Escobar Argueta	Josué Rodolfo Morales Castillo

C	Moises Eduardo Velasquez Oliva	David Augusto Maldonado Hurtarte
D	Herman Igor Veliz Linares	Esteban Humberto Valdez Ennati
E	Neftali de Jesus Calderon Mendez	Rodrigo Antonio Porón De León
F	William Estuardo Escobar Argueta	Ayeser Cristián Oxlaj Juárez
G	Edgar Francisco Rodas Robledo	Federico David Zet Pajoc

Cronograma de actividades: