

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA



VICERRECTORADO ACADÉMICO

FACULTAD DE INGENIERIA DE PRODUCCION Y SERVICIOS

DEPARTAMENTO ACADÉMICO DE INGENIERIA DE SISTEMAS E INFORMATICA

SÍLABO 2024 B

ASIGNATURA: CIENCIA DE LA COMPUTACIÓN I

1. INFORMACION ACADÉMICA

Periodo Académico:	2024-B	
Escuela Profesional:	Ciencia de la Computación	
Código de la Asignatura:	1701209	
Nombre de la Asignatura:	Ciencia de la Computación I	
Duración:	1 Semestre	
Condición:	Obligatorio Semestre II	
Número de Créditos:	5	
Número de Horas:	Teóricas:	2
	Prácticas:	2
	Laboratorio:	4
Prerrequisito:	1701106 Fundamentos de Computación	

2. INFORMACION ADMINISTRATIVA

Docente	Grado Académico	Dpto. Académico	Total de Horas	Horario/Actividad
Eliana María Adriazola Herrera	Bachiller en Ingeniería de Sistemas, Estudios de Maestría en Ingeniería de Software culminados.	Ingeniería de Sistemas e Informática.	20 horas	

3. FUNDAMENTACIÓN

El curso empieza con una revisión de los constructores fundamentales de programación, dando un especial énfasis a los tipos de datos estructurados, procesamiento de listas y archivos, técnicas de búsqueda y ordenación, para luego introducir al paradigma orientado a objetos (OOP), acentuando una apropiada formulación y abstracción del dominio del problema en el proceso de programación para la construcción de programas, focalizándose en la definición y uso de clases junto con los fundamentos del diseño orientado a objetos, encapsulación y ocultación de información, separación de comportamiento e implementación, clases, subclasses y herencia, polimorfismo, jerarquías de herencia, enlace dinámico y genericidad.

4. SUMILLA

- a. Estructuras de Control de Programas
- b. Estructuras de Datos Estáticas
- c. Manejo de la Memoria
- d. Estructuras de Datos Dinámicas
- e. Archivos
- f. Clases y Objetos
- g. Herencia
- h. Sobrecarga
- i. Polimorfismo
- j. Genericidad
- k. Plantillas
- l. Patrones de POO

5. COMPETENCIAS DE LA ASIGNATURA

- a. Desarrolla habilidades para la propuesta de soluciones algorítmicas mediante las técnicas estudiadas.
- b. Estudia los paradigmas y técnicas de programación orientada a objetos fundamentales.
- c. Analiza y comprende conocimientos teóricos del paradigma Orientado a Objetos y después los pone en práctica por medio de algún lenguaje Orientado a Objetos.
- d. Aplica los conocimientos teóricos y prácticos adquiridos durante el curso, proponiendo, diseñando y desarrollando alguna aplicación que permita evaluar el grado de conocimientos adquiridos.

- e. Estudia los algoritmos y estructuras de datos que resuelven los problemas de mayor frecuencia en las aplicaciones de computación.
- f. Desarrolla la capacidad para abstraer problemas de modo tal que se puedan crear estructuras y tipos de datos abstractos para darles solución; así como, analizar su representación en la memoria.

6. CONTENIDOS DE LA ASIGNATURA

HORAY FECHA	CONCEPTUAL	PROCEDIMENTAL	ACTITUDINAL	%
	<ul style="list-style-type: none"> - Conocer los paradigmas de programación fundamentales. - Conocer cómo ayuda el paradigma orientado a objetos a mejorar las capacidades de reuso de software. 	<ul style="list-style-type: none"> - Investigar los modelos de programación. - Comparar las ventajas del lenguaje de programación C/C++. 	<ul style="list-style-type: none"> - Participar voluntaria y activamente en clase. - Mostrar interés sobre la temática del curso. - Valorar los diversos modelos de programación. 	5%
	<ul style="list-style-type: none"> - Conocer las estructuras de control de programas: Estructuras de control condicionales. (PF1). - Conocer las estructuras de control de programas: Estructuras de control iterativas. (PF1) 	<ul style="list-style-type: none"> - Utilizar estructuras condicionales. - Identificar estructuras de control que harán posible la solución de problemas. - Aplicar estructuras condicionales. - Utilizar estructuras de control iterativas. - Aplicar las estructuras cíclicas o bucles. 	<ul style="list-style-type: none"> - Valorar el uso de las estructuras de control condicionales y cíclicas en la resolución de problemas. - Demuestra responsabilidad en la entrega oportuna de trabajos. 	10%
	<ul style="list-style-type: none"> - Describir los principios de definición de Funciones y Procedimientos y la técnica de recursividad. (PF2) - Conocer las estructuras de datos tipo Listas: (PF3, PF4) - Entender la importancia y uso de Listas Multidimensionales. 	<ul style="list-style-type: none"> - Resolver problemas con la técnica de recursividad. - Utilizar los conceptos de definición de funciones. - Utilizar estructuras de datos homogéneas en la resolución de problemas. - Usar las estructuras de datos homogéneas para la gestión de la data. 	<ul style="list-style-type: none"> - Proponer el uso de funciones en la solución de problemas. - Proponer el uso de estructuras de datos homogéneas para resolver problemas. - Demostrar responsabilidad en la entrega oportuna de trabajos. 	20%

	<ul style="list-style-type: none"> - Analizar problemas de matrices utilizando listas de listas. 			
	<ul style="list-style-type: none"> - Entender los conceptos de clase, atributo, operación, interfaz y objeto. - Entender el mecanismo de paso de mensajes. - Comprender los tipos de relación entre clases. - Entender el concepto de estado de un objeto. 	<ul style="list-style-type: none"> - Mejorar estructuras de programación básicas. - Proponer diversos métodos de solución a un problema, analiza y compara la eficiencia de cada método. 	<ul style="list-style-type: none"> - Proponer la creación de tipos de datos abstractos en la solución de problemas. - Demostrar responsabilidad en la entrega oportuna de trabajos. 	35%
	<ul style="list-style-type: none"> - Entender el mecanismo de herencia. - Saber discernir entre jerarquías de herencia seguras. - Comprender los costes de la herencia. - Saber decidir cuándo usar herencia y cuándo optar por composición. 	<ul style="list-style-type: none"> - Diseñar diversos métodos de solución a un problema - Analizar y comparar la eficiencia de cada método. 	<ul style="list-style-type: none"> - Proponer los tipos de relaciones correctos entre clases en la solución de problemas. - Demostrar responsabilidad en la entrega oportuna de trabajos. 	45%
	<ul style="list-style-type: none"> - Revisar los Métodos de ordenación: (AL1) <p>Métodos: Burbuja, InsertSort, QuickSort, Shell Sort y Ordenamiento basado en colas.</p>	<ul style="list-style-type: none"> - Utilizar los diversos métodos de ordenación. - Comparar la eficiencia de los métodos de ordenación. 	<ul style="list-style-type: none"> - Proponer el uso de métodos de ordenación y búsqueda eficientes. - Demuestra responsabilidad en la entrega oportuna de trabajos. 	50%
	PRIMER EXAMEN			
	<ul style="list-style-type: none"> - Conocer las funciones para el procesamiento de Cadenas de caracteres (Strings). 	<ul style="list-style-type: none"> - Resolver problemas con la ayuda del procesamiento de cadenas. - Programar las funciones más usadas del procesamiento de cadenas. 	<ul style="list-style-type: none"> - Programar las funciones de procesamiento de strings. - Demuestra responsabilidad en la entrega oportuna de trabajos. 	55%
	<ul style="list-style-type: none"> - Conocer las técnicas de escritura y lectura 	<ul style="list-style-type: none"> - Utilizar archivos de texto para el almacenamiento de 	<ul style="list-style-type: none"> - Proponer el uso de archivos para el 	65%

	de Archivos. (PF3)	<ul style="list-style-type: none"> - la data. - Resolver problemas con las técnicas de gestión de archivos. 	<ul style="list-style-type: none"> - almacenamiento persistente de la data. - Demuestra responsabilidad en la entrega oportuna de trabajos. 	
	<ul style="list-style-type: none"> - Entender el concepto de polimorfismo. - Entender la diferencia entre ligadura estática y ligadura dinámica en los lenguajes de programación. - Entender la relación a nivel de implementación entre herencia y polimorfismo. - Conocer el comportamiento típico de las pilas y colas. 	<ul style="list-style-type: none"> - Diseñar el funcionamiento de estructuras de datos tipo pilas y colas. - Resolver problemas con el uso de pilas y colas. 	<ul style="list-style-type: none"> - Proponer el uso de las estructuras tipo pilas y colas en la resolución de problemas. - Demuestra responsabilidad en la entrega oportuna de trabajos. 	75%
	<ul style="list-style-type: none"> - Saber identificar los distintos tipos de polimorfismo: sobrecarga, sobrescritura, variables polimórficas y genericidad. 	<ul style="list-style-type: none"> - Resolver problemas con los principios de programación orientada a objetos. - Utilizar las técnicas de polimorfismo en POO. - Diseñar soluciones que se adapten al modelo de POO. 	<ul style="list-style-type: none"> - Valorar los principios de la programación orientada a objetos. - Diseñar clases y objetos para la solución de problemas. - Demuestra responsabilidad en la entrega oportuna de trabajos. 	80%
	SEGUNDO EXAMEN			
	<ul style="list-style-type: none"> - Entender los mecanismos de gestión de errores. - Entender el concepto de persistencia. - Entender el concepto de concurrencia. 	<ul style="list-style-type: none"> - Resolver problemas con los principios de programación orientada a objetos. - Utilizar las técnicas de concurrencia. 	<ul style="list-style-type: none"> - Valorar los principios de la programación orientada a objetos. - Diseñar clases y objetos para la solución de problemas. - Demuestra responsabilidad en la entrega oportuna de trabajos. 	85%
	<ul style="list-style-type: none"> - Entender los Métodos de búsqueda: (AL1) - Búsqueda secuencial. - Búsqueda binaria. 	<ul style="list-style-type: none"> - Utilizar los distintos métodos de búsqueda. - Comparar la eficiencia de los métodos de búsqueda en listas. 	<ul style="list-style-type: none"> - Proponer el uso adecuado del método de búsqueda más eficiente. - Demuestra responsabilidad 	90%

	Análisis de los tiempos de búsqueda.		en la entrega oportuna de trabajos.	
	- Revisar el uso de diversas librerías gráficas.	- Utilizar librerías gráficas que apoyen en el entendimiento de las estructuras de control de programas. - Aplicar las librerías gráficas proporcionadas por el lenguaje.	- Proponer el uso de librerías gráficas que apoyan en el entendimiento del curso. - Demuestra responsabilidad en la entrega oportuna de trabajos.	95%
	- Conocer las técnicas de modificación de imágenes y sonidos mediante ciclos.	- Utilizar librerías que permitan controlar y modificar las características de una imagen. - Utilizar librerías que permitan controlar y modificar los sonidos. - Resolver problemas con las técnicas de modificación de audio e imagen.	- Valorar las técnicas de modificación de audio e imagen en la resolución problemas. - Demuestra responsabilidad en la entrega oportuna de trabajos.	100%
	TERCER EXAMEN			

7. ESTRATEGIAS DE ENSEÑANZA

Para el desarrollo del curso se ha previsto realizar las siguientes actividades:

- ✓ Clases magistrales.
- ✓ Clases prácticas.
- ✓ Exposiciones e intervenciones.
- ✓ Prácticas guiadas personales y en grupo.
- ✓ Trabajos de investigación en grupos.
- ✓ Prácticas en laboratorio de cómputo.

8. CRONOGRAMA ACADÉMICO

SEMANA	TEMA	PORCENTAJE DE AVANCE
	- Fundamentos de Programación. - Paradigmas de Programación.	5%
	- Constructores de Programación - Estructuras de Control de Programas	10%
	- Funciones y Procedimientos. - Punteros	15%
	- Principios de la Programación Orientada a Objetos.	25%

	- Herencia.	40%
	- Uso de librerías gráficas.	50%
	- Polimorfismo	65%
	- Genericidad	75%
	- Listas.	77%
	- Métodos de ordenación y búsqueda.	80%
	- Matrices.	83%
	- Archivos	85%
	- Estructuras de Datos.	90%
	- Mecanismos de Gestión de Errores	95%
	- Más sobre Programación Orientada a Objetos.	97%
	- Administración de la Memoria.	100%
	- Técnicas de modificación de imágenes y sonidos mediante ciclos.	

9. ESTRATEGIAS DE EVALUACIÓN

Se emplearan las siguientes estrategias de evaluación:

- ✓ Exámenes escritos.
- ✓ Trabajos de investigación.
- ✓ Desarrollo de prácticas.
- ✓ Exámenes de laboratorio.

Nº	Fecha Exámenes Teoría	Fecha Exámenes Practica	Otros
1o	02-10-2024	02-10-2024	
2o	13-11-2024	13-11-2024	
3o	18-12-2024	18-12-2024	

10. REQUISITOS DE APROBACIÓN

El sistema de evaluación es el siguiente:

- | | | |
|-------------------------|------|------|
| • Primer examen | peso | 12% |
| • Segundo examen | peso | 12% |
| • Tercer examen | peso | 16% |
| • Evaluación Continua 1 | peso | 18% |
| • Evaluación Continua 2 | peso | 18% |
| • Evaluación Continua 3 | peso | 24% |
| | | 100% |

11. BIBLIOGRAFÍA

Obligatoria:

- C++ How to Program (10th Edition) 10th Edition, Paul J. Deitel, Harvey Deitel.
- The C++ Programming Language, Stroustrup Bjarne.

Sugerida:

- Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Grady Booch.
- Thinking in C++. 2nd Edition. Bruce Eckel. Libro electrónico gratuito.
<http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html>

FECHA: AREQUIPA, 02 DE SEPTIEMBRE DEL 2024

FIRMA DEL DOCENTE

ELIANA M. ADRIAZOLA HERRERA