

# CIENCIA DE LA COMPUTACIÓN

---

**Programación Orientada a Objetos I**

**4 CRÉDITOS**



# ÍNDICE

<b>1. Asignatura</b>	<b>3</b>
<b>2. Datos generales</b>	<b>3</b>
<b>3. Profesores</b>	<b>3</b>
3.1 Profesor coordinador del curso	3
3.2 Profesor(es) instructor(es) del curso	3
<b>4. Introducción al curso</b>	<b>4</b>
<b>5. Objetivos</b>	<b>4</b>
<b>6. Competencias</b>	<b>5</b>
<b>7. Resultados de aprendizaje</b>	<b>6</b>
<b>8. Temas</b>	<b>6</b>
<b>9. Plan de trabajo</b>	<b>7</b>
<b>10. Sistema de evaluación</b>	<b>8</b>
<b>11. Sesiones de apoyo o tutorías</b>	<b>8</b>
<b>12. Referencias Bibliográficas</b>	<b>9</b>

# UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

## SILABO 2020-2

### 1. ASIGNATURA

CS1102 - Programación Orientada a Objetos I

### 2. DATOS GENERALES

**2.1. Créditos:** cuatro (4) créditos

**2.2. Horas de teoría:** dos (2) semanales

**2.3. Horas de práctica:** cuatro (4) semanales

**2.4. Duración del período:** dieciséis (16) semanas

**2.5. Condición:**

- Obligatorio para Ciencia de la Computación

**2.6. Modalidad:** Virtual

**2.7. Requisitos**

CS1100. Introducción a la Ciencia de la Computación (1° Sem)

### 3. PROFESORES

#### 3.1. Profesor coordinador del curso

José Antonio Fiestas Iquira ([jfiestas@utec.edu.pe](mailto:jfiestas@utec.edu.pe))

Horario de atención: Lunes de 5 pm - 6 pm

#### 3.2. Profesor(es) instructor(es) del curso

Heider Sanchez ([hsanchez@utec.edu.pe](mailto:hsanchez@utec.edu.pe))

Horario de atención: previa coordinación con el profesor

Maria Hilda Bermejo Rios ([mbermejo@utec.edu.pe](mailto:mbermejo@utec.edu.pe))

Horario de atención: previa coordinación con el profesor

Ruben Rivas Medina ([rrivas@utec.edu.pe](mailto:rrivas@utec.edu.pe))

Horario de atención: previa coordinación con el profesor

Henry Giovanni Gallegos Velgara ([hgallegos@utec.edu.pe](mailto:hgallegos@utec.edu.pe))

Horario de atención: previa coordinación con el profesor

Marvin Abisrro Zarate ([mabisrro@utec.edu.pe](mailto:mabisrro@utec.edu.pe))

Horario de atención: previa coordinación con el profesor

Pedro Nelson Shiguihara Juárez ([pshiguihara@utec.edu.pe](mailto:pshiguihara@utec.edu.pe))

Horario de atención: previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este es el segundo curso en la secuencia de los cursos introductorios a la Ciencia de la Computación. El curso introducirá a los participantes en los fundamentos de la programación orientada a objetos. Se utiliza el lenguaje C++, el cual se ha establecido como muy estable, portable, y creado tanto para expertos como para principiantes. De suma importancia en Ciencia de la Computación, es el diseño del algoritmo y la programación. El estudiante primero crea modelos algorítmicos y/o de datos, y luego debe programarlos para probarlos y validarlos. La programación orientada a objetos enfoca el diseño de la solución al objeto, que es el ente manejado por el algoritmo de solución al problema enfocado. Este método brinda una serie de ventajas de optimización y estabilidad del código, lo que permite al estudiante, obtener un producto de mayor calidad y valor en el mercado computacional.

## 5. OBJETIVOS

Sesión 1:

- Comprender el diseño de un algoritmo y la estructura de un programa en C++
- Conocer y usar tipos de variables operadores
- Conocer y usar estructuras de control

Sesión 2:

- Entender el concepto de variables globales
- Comprender el concepto de transferencia por valor y por referencia
- Usar funciones recursivas y compararlas con no-recursivas

Sesión 3:

- Comprender y usar la memoria de forma optimizada con punteros
- Utilizar arreglos dinámicos en un código en C++
- Comprender la definición y uso de punteros

Sesión 4:

- Usar vectores en sus variaciones para optimizar el desempeño de un código en C++

Sesión 5:

- Comprender el proceso de abstracción y encapsulamiento y la relación clase-objeto
- Utilizar funciones de definición de objetos de clase
- Optimizar el código con objetos y arreglos dinámicos

Sesión 6:

- Comprender y utilizar los conceptos de asociación, composición y agregación

- Usar herencia y funciones virtuales en el diseño de un algoritmo en C++

Sesión 7:

- Utilizar polimorfismo en casos de herencia para mejorar organización y legibilidad de código

Sesión 8:

- Comprender las ventajas de utilización de sobrecarga de operadores
- Aplicar este concepto en casos diversos de programación

Sesión 9:

- Usar plantillas para reutilizar un código fuente de programación
- Conocer y utilizar archivos para optimizar el manejo de datos de entrada y salida en un código de programación

## 6. COMPETENCIAS

### Para Ciencia de la Computación:

a2. Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina.

El estudiante responde a una evaluación teórica (exámenes) para medir esta competencia.

a3. Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución.

El estudiante desarrolla el proyecto del curso para medir esta competencia.

b3. Utilizar técnicas y herramientas de programación para la solución de problemas planteados.

El estudiante desarrolla ejercicios de Programación en Laboratorio para medir esta competencia.

### Para Ingeniería:

a3. Capacidad de aplicar conocimientos de matemáticas.

El estudiante responde a una evaluación teórica (exámenes) para medir esta competencia

c2. Capacidad para diseñar un sistema, un componente o un proceso. para satisfacer las necesidades deseadas dentro de restricciones realistas.

El estudiante desarrolla el proyecto del curso para medir esta competencia.

## 7. RESULTADOS DE APRENDIZAJE

Al final del curso de Programación orientada a objetos 1 se espera:

**RA1.** Que el estudiante sea capaz de conocer los fundamentos del paradigma de programación orientada a objetos, permitiendo asimilar los conceptos necesarios para desarrollar sistemas de información.

**RA2.** Que el estudiante sea capaz de optimizar un código a través del uso de la programación orientada a objetos con el fin de lograr un mejor performance durante la ejecución del software

## 8. TEMAS

1. Introducción al diseño de algoritmos y programación en C++
  - 1.1 Diseño de un algoritmo y estructura de un programa en C++ (rol de los algoritmos en el proceso de solución de problemas)
  - 1.2 Variables y tipos de datos primitivos (ej., números, caracteres, booleanos) . Tipos de datos y operadores
  - 1.3 Estructuras de control selectivas y repetitivas
2. Funciones y recursividad
  - 2.1 Scope o ámbito: variables globales
  - 2.2 Paso de funciones y parámetros. Transferencia por valor y por referencia
  - 2.3 Funciones matemáticas iterativas y recursivas
- 3: Punteros
  - 3.1 Definición y manejo de punteros
  - 3.2 Análisis de punteros en código
  - 3.4 Manejo de memoria (dinámica)
  - 3.5 Arreglos dinámicos (arrays, vectores, matrices)
4. Vectores
  - 4.1 Definición y construcción
  - 4.2 Operadores básicos
- 5: Programación orientada a objetos
  - 5.1 Proceso de abstracción
  - 5.2 Relación clase-objeto
  - 5.3 Métodos de acceso y modificación (getters y setters)
  - 5.4 Constructores y destructores
  - 5.5 Objetos dinámicos
  - 5.6 Arreglos de objetos
6. Relaciones entre clases
  - 6.1 Asociación
  - 6.2 Composición
  - 6.3 Agregación
  - 6.4 Herencia (funciones virtuales)
7. Polimorfismo
  - 7.1 Polimorfismo paramétrico o genérico.
  - 7.2 Polimorfismo de inclusión o herencia.
  - 7.3 Herencia de tipos con Override
8. Sobrecarga de operadores
  - 8.1 Funciones amigas
  - 8.2 Sobrecarga de operadores
  - 8.3 Casos particulares de sobrecarga
9. Templates y archivos
  - 9.1 Definición de plantillas (templates)
  - 9.2 Archivos. Definición y modos de uso

## 9. PLAN DE TRABAJO

### 9.1 Metodología

La metodología del curso se enfoca en clases magistrales y se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar con el profesor y con los otros estudiantes.

### 9.2 Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con estrategias didácticas que permitan a los estudiantes interiorizar los conceptos.

### 9.3 Sesiones de Laboratorio:

Para verificar que los estudiantes hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, se realizan actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Para fomentar la participación individual y en equipo se proponen exposiciones individuales o grupales para presentar sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

## 10. SISTEMA DE EVALUACIÓN

	TEORÍA	PRÁCTICA Y/O LABORATORIO
<b>EVALUACIÓN</b>  *La ponderación de la evaluación se hará si ambas partes están aprobadas	1 Exámen Teórico (E) (35%)	Evaluacion Continua 1 (C1) Semanas 1-8 (5%) Evaluacion Continua 2 (C2) Semanas 9-16 (5%) 3 Practicas Calificadas (PC1, PC2, PC3) (15% c/u, total 45%) Proyecto (10%) (P)
	35%	65%

	<b>100%</b>
--	-------------

Las rúbricas que permitirán medir las actividades más significativas del curso y que, además se relacionan con la evaluación de las competencias del estudiante son: [enlace](#)

## 11. SESIONES DE APOYO O TUTORÍAS

Este apartado permite formalizar los espacios de apoyo a los estudiantes y que éstos tengan la atención NECESARIA y el tiempo disponible para presentar sus dudas y consultas acerca del curso:

Semana	Fecha/ Hora	Tema a tratar	Objetivos de la sesión
3	15/09/2020 2:00 PM – 3:00 PM	Inicio de ciclo	Aclarar dudas sobre evaluaciones, rubricas y laboratorios
5	29/09/2020 2:00 PM – 3:00 PM	Avance Teoría-Laboratorio	Evaluar concordancia entre clases Teóricas y Laboratorio
7	13/10/2020 2:00 PM – 3:00 PM	Proyectos	Evaluación de avance de proyectos y consultas al respecto
9	27/10/2020 2:00 PM – 3:00 PM	Avance Teoría-Laboratorio	Evaluar concordancia entre clases Teóricas y Laboratorio
11	10/11/2020 2:00 PM – 3:00 PM	Proyectos	Evaluación de avance de proyectos y consultas al respecto
13	24/11/2020 2:00 PM – 3:00 PM	Avance Teoría-Laboratorio	Evaluar concordancia entre clases Teóricas y Laboratorio
15	08/12/2020 2:00 PM – 3:00 PM	Cierre de ciclo	Evaluaciones finales y calificación

## 12. REFERENCIAS BIBLIOGRÁFICAS

- Deitel. P.J and Deitel. H.M. (2016) *C++ How to Program*, Prentice Hall
- Stroustrup, Bjarne (2013) *The C++ Programming Language*. 4th. Addison-Wesley.

- Eckel, Bruce, 2000, *Thinking in C++*, Vol. 1: *Introduction to Standard C++*, 2nd Edition, Prentice Hall