

PRONTUARIO

TÍTULO:	Análisis comparativo de lenguajes de programación
CODIFICACIÓN:	CCO 215
PRERREQUISITO:	INF 201
CRÉDITOS:	3 créditos 45 horas contacto

DESCRIPCIÓN

Estudio y análisis comparativo de distintos paradigmas, estructuras y técnicas de programación. Se estudia la sintaxis, semántica e implantación de lenguajes de programación. Se requieren proyectos independientes de programación. El curso tiene un componente en línea.

JUSTIFICACIÓN

Los lenguajes de programación son las herramientas más importantes para un programador. Si los lenguajes de programación son examinados cuidadosamente, se pueden encontrar semejanzas entre unos y otros, aun más de lo que su forma externa pueda reflejar. El estudio de los conceptos básicos de los lenguajes de programación ayudará al programador no sólo a aprender la estructura de varios lenguajes sino a hacer un uso efectivo y eficiente de los mismos.

COMPETENCIAS

El curso desarrolla en el o la estudiante las siguientes competencias:

Competencias transversales

1. Cuestionamiento crítico (Nivel II)

OBJETIVOS

Al finalizar el curso el o la estudiante será capaz de:

1. Conocer las características de los lenguajes de programación.
2. Conocer los elementos de la sintaxis de un lenguaje de programación.
3. Utilizar diferentes paradigmas de programación en la solución de problemas.
4. Comparar y analizar diferentes lenguajes de programación.

5. Reconocer la importancia del desarrollo histórico de los lenguajes de programación.

CONTENIDO

I. Principios de diseño

A. Elementos de un lenguaje

1. Definición de lenguaje de programación
2. Diseño de lenguaje
3. Proceso de programación
4. Criterios para evaluar y comparar lenguajes
5. Historia de los lenguajes de programación

B. Proceso de traducción

1. Análisis léxico, sintáctico y semántico
2. Generación de código y optimización de código
3. Compilación vs interpretación

II. Sintaxis

A. Estructura léxica

1. Gramáticas
2. Backus-Naur Form (BNF)
3. Derivación canónica, parse trees
4. Gramáticas ambiguas

III. Semántica

A. Manejo de nombres, localizaciones y valores

1. Declaraciones
2. Asignaciones
3. Expresiones
4. Binding
5. Verificación de tipos
6. Equivalencia de tipos
7. Alcance

B. Manejo de estructuras de control

1. Estructuras condicionales
2. Estructuras iterativas

- C. Manejo de memoria
 - 1. Arreglos
 - 2. Apuntadores
 - 3. Estructuras
 - 4. Sub-programas y parámetros
- IV. Paradigmas de programación
 - A. Programación orientada a objetos
 - 1. ADT
 - 2. Objetos
 - 3. Herencia
 - 4. Clases
 - B. Programación funcional
 - C. Programación lógica
- V. Manejo de excepciones
 - A. Especificaciones
 - B. Implantación

METODOLOGÍA

Se recomiendan las siguientes estrategias de la metodología de aprendizaje activo:

ESTRATEGIAS INSTRUCCIONALES

- Conferencia
- Discusión de temas
- Ejercicios y problemas de programación
- Uso de recursos disponibles en la Web
- Informes orales

RECURSOS

Acceso a la Internet

EVALUACIÓN

Exámenes parciales	40%
Programas y tareas	40%
Trabajo final	<u>20%</u>
Total	100%

AVALÚO DEL APRENDIZAJE

Se aplica la rúbrica de avalúo institucional a la actividad central del curso.

BIBLIOGRAFÍA

TEXTO

Sebesta, R.W. (2016). Concepts of Programming Languages (11th edition). Boston: Pearson.

REFERENCIAS

Abelson, H., Sussman, G. J., & Sussman, J. (1996). Structure and Interpretation of Computer Programs (2nd ed.). The MIT Press.

Alomari, Z., Halimi, O. E., Sivaprasad, K., & Pandit, C. (2015). Comparative studies of six programming languages.

Douglas Crockford. (2008). JavaScript: The good parts. O'Reilly Media.

Farooq, M. S., Khan, S. A., Ahmad, F., Islam, S., & Abid, A. (2014). An evaluation framework and comparative analysis of the widely used first programming languages. Plos ONE, 9(2), 1-25. doi:10.1371/journal.pone.0088941

Gonçalves, J. (2015). Functional Programming should be your #1 priority for 2015. Recuperado de <https://medium.com/@jugoncalves/functional-programming-should-be-your-1-priority-for-2015-47dd4641d6b9#.ckwdwo3h2>

HaskellWiki. (s.f.). Functional programming. Recuperado el 8 de marzo de 2016 de https://wiki.haskell.org/Functional_programming

Haverbeke, M. (2014). Eloquent JavaScript: A modern introduction to programming ed.). San Francisco: No Starch Press. (2nd.

Naim, R., Nizam, M. F., Hanamasagar, S., Nouredine, J., & Miladinova, M. (2010). Comparative studies of 10 programming languages within 10 diverse criteria - a Team 10 COMP6411-S10 Term Report.

Tucker, A. B., & Noonan, R. (2006). Programming languages. Boston: McGraw-Hill Higher Education.

Puede encontrar más recursos de información relacionados a los temas del curso en la página de la biblioteca <http://biblioteca.sagrado.edu/>

Las bases de datos electrónicas a las cuales la Biblioteca Madre María Teresa Guevara está suscrita directamente y a través del Consorcio COBIMET, incluyen, documentos, artículos de revistas y periódicos y otros recursos de información relacionados con los temas del curso. Al utilizarlas siga los siguientes pasos:

Para **acceder desde cualquier lugar en la Universidad**

- escriba la dirección <http://biblioteca.sagrado.edu/>,
- seleccione **Biblioteca Virtual** y aparecerá la página en donde podrá acceder a las bases de datos, por disciplina o en orden alfabético.

Para **acceder fuera de la Universidad**

- escriba la dirección <http://biblioteca.sagrado.edu/>,
- seleccione **Biblioteca Virtual** y aparecerá la página en donde podrá acceder a las bases de datos, por disciplina o en orden alfabético.
- escriba el nombre del usuario y la contraseña (El nombre de usuario y la contraseña, los solicita personalmente en la Biblioteca)

ACOMODO RAZONABLE

Para obtener información detallada del proceso y la documentación requerida, debe visitar la oficina correspondiente. Para garantizar igualdad de condiciones, en cumplimiento de la Ley ADA (1990) y el Acta de Rehabilitación (1973), según enmendada, todo estudiante que necesite servicios de acomodo razonable o asistencia especial deberá completar el proceso establecido por la Vicepresidencia de Asuntos Académicos.

INTEGRIDAD ACADÉMICA

Esta política aplica a todo estudiante matriculado en la Universidad del Sagrado Corazón para tomar cursos con o sin crédito académico. Una falta de integridad académica es todo acto u omisión que no demuestre la honestidad, transparencia y responsabilidad que debe caracterizar toda actividad académica. Todo estudiante que falte a la política de honradez, fraude y plagio se expone a las siguientes sanciones: recibirá nota de cero en la evaluación y/o repetición del trabajo en el seminario, nota de F(*) en el seminario: suspensión o expulsión según se establece en el documento de Política de Integridad Académica con fecha de efectividad de noviembre 2022.

CURSOS DE INVESTIGACIÓN

“Este curso puede requerir que los estudiantes practiquen tareas relacionadas al proceso de investigación, tales como: toma de consentimiento o asentimiento informado, administración de instrumentos, realización de entrevistas, observaciones o grupos focales, entre otros. Estas tareas son parte de un ejercicio académico y no se utilizará la información recopilada para compartirla con terceros o divulgar en otros escenarios que no sean el salón de clases junto al profesor que enseña el curso. Todo estudiante que vaya a interactuar con sujetos humanos como parte de su práctica en investigación tiene que estar certificado en ética con sujetos humanos en la investigación por el *Collaborative Institutional Training Initiative (CITI Program)*, al igual que su profesor”(Los cursos de investigación deben incluir este texto)