

LENGUAJES DE PROGRAMACIÓN

Semestre Enero-Mayo de 2018

TAREA 1: Conceptos sobre Lenguajes de Programación

Forma de trabajo: Equipo de 2 integrantes

Forma de entrega: Examen individual en clase y documento del equipo en Blackboard.

Fecha de entrega: 1ª parte (durante la clase del martes 16 de enero de 2018) y 2ª parte (martes 23 de enero de 2018)

PRIMERA PARTE: Repaso y evaluación de conceptos previos (Individual - 30 puntos)

Repasa los conceptos sobre teoría de lenguajes vistos en tu clase de Matemáticas Computacionales, incluyendo los temas de: Autómatas Finitos Deterministas y No deterministas, Expresiones Regulares, Gramáticas Regulares, Gramáticas Libres de Contexto, Autómatas de Pila, Analizadores Sintácticos Descendentes y Ascendentes, y Máquinas de Turing. Asegúrate de tener claras las ventajas y desventajas de cada una de estas herramientas para la representación de lenguajes. Este punto de la tarea será evaluado a través de un examen rápido de diagnóstico en la fecha establecida.

SEGUNDA PARTE: Lectura y reflexión sobre conceptos básicos (En equipo)

Crear el documento Word **M_Tarea1.docx** (donde sustituyan la M con sus matrículas) con los resúmenes solicitados abajo en páginas con márgenes normales y Font Arial de tamaño 12. Resúmenes que no cubran todo lo leído o no cumplan con la extensión solicitada (+/- un cuarto de página) serán penalizados. El documento debe ser subido a Blackboard por uno solo de los integrantes del equipo.

PROBLEMA 1 (20 puntos)

Leer las secciones 1.1 *What is a Programming Language?* 1.2 *Abstractions in Programming Languages* y 1.3 *Computational Paradigms* del capítulo 1 *Introduction* del libro *Programming Languages: Principles and Practice, 2nd. Edition* de Kenneth Loudon que podrás encontrar en la sección *Documentos del Curso* en Blackboard. Escribir un resumen de **3 páginas** sobre lo que leyeron.

PROBLEMA 2 (20 puntos)

La siguiente página es una interesante referencia que muestra la popularidad y la tendencia de los lenguajes de programación: <http://www.tiobe.com/tpci.htm>. Después de revisar esta página, investiguen en internet, **clasifiquen y reporten los primeros 50 lenguajes de programación que ahí aparecen dentro de las 4 clasificaciones vistas en clase** (por generación, por nivel de abstracción, por la forma de expresar la solución a un problema y por paradigma). Identifiquen SOLO la categoría principal en la que cae cada lenguaje en cada clasificación, por ejemplo, aunque Scheme es multi-paradigma, su paradigma principal es el funcional.

PROBLEMA 3 (30 puntos)

Un concepto muy relevante en la teoría e implementación de lenguajes de programación es el que tiene que ver con su sistema de tipos. Investiguen y contesten las siguientes preguntas incluyendo el resultado en el documento de la tarea. Las **descripciones de los ejemplos deben explicarse** para ilustrar los conceptos:

1. ¿Qué es y para que se utiliza un tipo de dato? Describir un ejemplo de tipo de dato.
2. ¿Qué es y para que se utiliza un sistema de tipos? Describir un ejemplo de uso de un sistema de tipos.
3. ¿Cuáles son los lenguajes de programación fuertemente tipados? Mencionar dos lenguajes de programación fuertemente tipados (nombre).

4. ¿Cuáles son los lenguajes de programación débilmente tipados? Mencionar dos lenguajes de programación débilmente tipados (nombre).
5. ¿En qué consiste el chequeo estático de tipos? Describir un ejemplo de cómo trabaja este tipo de chequeo.
6. ¿En qué consiste el chequeo dinámico de tipos? Describir un ejemplo de cómo trabaja este tipo de chequeo.
7. ¿En qué consiste el tipado seguro en lenguajes de programación? Describir un ejemplo de tipado seguro en un lenguaje de programación.
8. ¿Cuál es la diferencia entre equivalencia de tipos, compatibilidad de tipos, e inferencia de tipos en un sistema de tipos?
9. ¿Qué significa coerción de tipos? Describir un ejemplo donde se aplique.
10. ¿Qué es un sistema de tipos nominal? Describir un ejemplo de cómo trabaja un lenguaje con este tipo de sistema de tipos (que no sea el de Wikipedia)
11. ¿Qué es un sistema de tipos estructural? Describir un ejemplo de cómo trabaja un lenguaje con este tipo de sistema de tipos (que no sea el de Wikipedia)