

# Información del curso

Pedro O. Pérez M., PhD.

Diseño de compiladores  
Tecnológico de Monterrey

*pperezm@tec.mx*

07-2020

## Información del profesor

Información del profesor

## Información del curso

Intenciones educativas

Objetivos generales

Metodología

Evaluación

Normas de clase

Bibliografía

## Información del profesor

- ▶ Pedro Oscar Pérez Murueta
  - ▶ ISC Mayo 1994
  - ▶ MTI Mayo 2002
  - ▶ DCC Diciembre 2019
- ▶ Correo: [pperezm@tec.mx](mailto:pperezm@tec.mx)
- ▶ Oficina: Edificio 2, Piso 3
- ▶ Horario de asesoría: Se encuentra en la puerta de mi oficina.



## Intenciones educativas

Curso de nivel avanzado en el área de Computación que proporciona a los estudiantes las herramientas necesarias para diseñar nuevos lenguajes de programación. Requiere de conocimientos previos de matemáticas computacionales, análisis de algoritmos, lenguajes de programación. Como resultado del aprendizaje, el estudiante implementará un nuevo lenguaje de programación básico, de propósito específico. La aplicación particular del lenguaje pudiera variar dependiendo de las necesidades y las tecnologías disponibles.

# Objetivos generales

Al finalizar el curso, el alumno será capaz de diseñar e implementar un nuevo lenguaje de programación.

# Metodología

Para alcanzar los objetivos que persigue la materia hemos desarrollado una estrategia organizada de la siguiente manera:

- ▶ **Autoestudio:** En algunas sesiones será necesario que hagas trabajo por ti mismo. En general, este trabajo consistirá en leer un capítulo de un libro o sitio Web, un conjunto de ejercicios o cualquier otro material proporcionado por el profesor.
- ▶ **Sesiones de clase:** Durante la sesión de clase se presentarán algunos conceptos importantes tomados del autoestudio. Debes usar este tiempo para resolver cualquier duda sobre el material o tareas visto en el autoestudio.

- ▶ **Actividades en clase:** Durante estas actividades se resolverán algunos ejercicios relacionados con la materia. El objetivo de los mismos es reforzar el conocimiento teórico presentado anteriormente.
- ▶ **Exámenes de tercio:** Las fechas de estos exámenes los encontrarás a continuación.
- ▶ **Examen final:** El objetivo principal de este curso es construir un compilador individualmente. Para lograr este objetivo, habrá dos revisiones parciales en las que entregarás las partes del compilador que se han revisado durante esos periodos. Al final del curso, presentarás el compilador completo y la documentación correspondiente, así como los casos de prueba.

## Fechas de exámenes:

- ▶ Primer examen: Semana 5, 10 de septiembre.
- ▶ Segundo examen: Semana 10, 15 de octubre.
- ▶ Tercer examen: Semana 15, 19 de noviembre.

## Puntos a recalcar:

- ▶ **Entregas parciales:** Si el proyecto no compila o se ejecuta, tiene 20 puntos. Se entrega en la fecha designada y es individual.



# Evaluación

Evaluación parcial		Evaluación final	
Actividades de clase	20 %	Actividades de clase	20 %
Entrega parcial	20 %	Proyecto final	20 %
Exámenes de tercio	60 %	Exámenes de tercio	30 %
		Examen final	20 %
		Foros	10 %

# Normas de clase

## Exámenes

- ▶ Los exámenes podrán ser presentados solamente en la fecha estipulada. El no presentar un examen implica una calificación de NP (No Presentó).
- ▶ El cambio de fecha de algún examen parcial deberá realizarse, a petición de los estudiantes, durante las dos primeras semanas de clase. Éste se hará sólo si se cuenta con el consenso del grupo y del profesor.

## Asistencia a clases

En lo que respecta a esta clase:

- ▶ La sesión de clase inicia 5 minutos después del horario establecido (10:05). Si no estás al inicio de la misma, se considerará que no asististe a esa sesión. Asimismo, también se considera inasistencia si te retiras, sin permiso del profesor, antes de terminar la sesión de clase.
- ▶ No podrás acreditar, bajo ningún concepto, las actividades (tareas y/o exámenes) de las sesiones a las cuales no hayas asistido. Además, será tu responsabilidad estudiar el material visto en esas sesiones.

## Tareas y Proyectos

- ▶ Toda tarea y/o proyecto tendrá su fecha y horario de entrega que es inamovible. Vencido el término de entrega no se recibirán tareas y/o proyectos.
- ▶ Todas las tareas son individuales a menos que explícitamente se pida trabajar en grupo.

## Redacción y Organización

- ▶ La mala redacción, organización y ortografía en la elaboración de tareas, proyectos, presentaciones y exámenes, será causa de penalización en la calificación correspondiente.

## Calificaciones

- ▶ Las calificaciones parciales y final se expresan en escala de uno a cien.
- ▶ La calificación mínima aprobatoria es 70 (SETENTA).

## Faltas a la Integridad Académica en Tareas, Proyectos o Exámenes

- ▶ Las faltas a la integridad académica, como la copia o tentativa de copia en cualquier tipo de examen o actividad de aprendizaje; el plagio parcial o total; facilitar alguna actividad o material para que sea copiada y/o presentada como propia; la suplantación de identidad; falsear información; alterar documentos académicos; vender o comprar exámenes o distribuirlos mediante cualquier modalidad; hurtar información o intentar sobornar a un profesor o cualquier colaborador de la institución; entre otras acciones más son consideradas faltas grave. Cuando un alumno cometa un acto contra la integridad académica, se le asignará una calificación reprobatoria a la actividad, examen, período parcial o final. La calificación reprobatoria asignada por el profesor será inapelable, y a esta sanción se sumarán las otras posibles que determine el Comité de Integridad Académica de Campus. Esto tal como lo indica el Reglamento Académico en su CAPÍTULO IX: Faltas a la integridad académica.

## Baja de materias

- ▶ Artículo 4.10: Los alumnos podrán darse de baja en una o varias materias inscritas durante el periodo académico semestral, trimestral o intensivo, **antes del último día de clases, conforme lo establezca el Calendario Escolar correspondiente.** Las materias dadas de baja no se registrarán como reprobadas. En el caso de que el alumno solicite la baja de todas sus materias, deberá pasar nuevamente por un proceso de admisión, en el que se tomará en cuenta su historial académico para la decisión. El proceso relacionado con la baja de las materias se llevará a cabo de acuerdo con las políticas administrativas definidas para ese fin.

# Bibliografía

- ▶ ULLMAN, Jeffrey D.; AHO, Alfred V. Principles of compiler design. Reading: Addison Wesley, 1977.
- ▶ GRUNE, Dick, et al. Modern compiler design. Springer Science & Business Media, 2012.
- ▶ ANDREW, W. Appel; JENS, P. Modern compiler implementation in Java. En ISBN 0-521-58388-8. Cambridge University Press, 2002.
- ▶ APPEL, Andrew W. Modern compiler implementation in C. Cambridge university press, 2004.



**SCAN ME**

<https://itesm.zoom.us/my/pperezm>



**SCAN ME**

<https://shorturl.at/ertE1>



**SCAN ME**

<https://shorturl.at/erwBG>



## Github



[https://github.com/Manchas2k4/  
compilers](https://github.com/Manchas2k4/compilers)

## Remind



[https:  
//www.remind.com/join/k7hgk2](https://www.remind.com/join/k7hgk2)

## ¿Qué es el Club de Programación Competitiva?

- ▶ El Club de Programación Competitiva se trata de hacer problemas ridículamente difíciles dentro de plazos injustamente cortos. Nuestro enfoque son los concursos como ACM ICPC, VCPC, Google Codejam y Facebook Hacker Cup.
- ▶ Tenemos reuniones quincenales en las que practicamos la solución de problemas, discutimos algoritmos, lloramos por las partes que no entendimos, entendemos otras y celebramos con pizza gratis.

# ¿Qué es el Club de Programación Competitiva?

- ▶ ¿Dónde? Salón por definir.
- ▶ ¿Cuándo? Los viernes, cada quince días, a partir de 28 de febrero.
- ▶ ¿En qué horario? 2 a 4 pm.

## ¿Cuál es la oferta?

- ▶ Por cada problema B o C que resuelvas en el Club de Programación Competitiva, tendrás un punto extra en tu calificación final... con las siguientes condiciones:
  - ▶ Deberás explicar el problema resuelto, a más tardar, antes de que empiece el siguiente concurso.
  - ▶ **Los puntos extras se aplicaran, SI Y SÓLO SI APRUEBAS LA MATERIA.**