

**DISCLAIMER:** This handout is written in Spanish (except for this first paragraph, obviously). The university does not allow important “contractual” (policy) information between the professor and the student concerning evaluation, dates, etc. to be published in any language but Spanish... which makes sense. In any case, I apologize for this inconvenience. Who else cares about this stuff anyway?! ■

## Información Importante

Este handout contiene información importante sobre el curso. La mayor parte de las secciones de este documento serán útiles a lo largo de todo el curso, así es que por favor mantenganlo cerca. Los puntos más importantes a prestar atención *ahora mismo* son:

1. En esta primera clase definiremos los horarios de laboratorio. Asegúrense de anotarse en uno u otro horario.
2. Anoten en sus calendarios las fechas de los certámenes y assignments, y asegúrense de tener esos horarios libres.
3. Por favor, lean cuidadosamente la política de colaboración con respecto a las tareas.
4. Por favor lean cuidadosamente la política de corrección de assignments. En particular, fíjense que si no aprueban los assignments o los certámenes con una nota promedio  $\geq 4$ , **NO APRUEBAN EL CURSO.**
5. Por favor, regístrense en Piazza: [piazza.com/udec.cl/spring2015/503208](http://piazza.com/udec.cl/spring2015/503208)

## 1 Staff

El profesor de este curso es Leo Ferres. Por favor revisen el sistema Infoda y Piazza para los nombres e información de contacto de profesor y alumnos ayudantes.

El website del curso es:

[http://www.leoferres.info/teaching/2015\\_1\\_503208.html](http://www.leoferres.info/teaching/2015_1_503208.html)

Si necesitan alguna clase de ayuda, lo mejor es postear la pregunta en Piazza.

[piazza.com/udec.cl/spring2015/503208/home](http://piazza.com/udec.cl/spring2015/503208/home)

Así, cualquiera de nosotros (el profesor, los ayudantes o inclusive otros compañeros) pueden responder, haciendo que la respuesta llegue antes. A menos que necesiten una respuesta por un asunto privado, usen Piazza. Para todo email que llegue al staff en privado y que concierna a todo el grupo, se le pedirá al autor del email que lo publique en Piazza.

Los ayudantes son Aníbal, Diego, y Meraioth. Ellos tienen varias obligaciones durante el curso:

---

<sup>1</sup>This set of rules has been modified to fit this particular course from the original, created by E. Demain for his 6.046 course at MIT.

1. son responsables de los laboratorios. Al final del curso, ellos me pasan una nota de participación de cada uno de ustedes en dichos laboratorios
2. son los que corrigen los assignments y el certamen de laboratorio
3. son los que cuidan los certámenes
4. tienen un horario de consultas

Los ayudantes de este curso son personas muy inteligentes y saben mucho. Saquen provecho a esto.

## 2 Temas

En esta clase abordaremos diferentes temas de programación procedural. Algunos de estos temas en general son, entre otros:

- Arquitectura y representación de datos
- Variables
- Operadores y expresiones
- Conversión de tipos de datos y orden de evaluación
- Condicionales, interacción
- Funciones y estructura de programa
- Vectores
- Strings
- Punteros
- Structs
- Memoria dinámica
- Entrada/Salida

## 3 Prerrequisitos

No hay pre-requisitos para 503208.

A veces, estudiantes de otras carreras vienen a tomar este curso. Esos estudiantes tienen que venir a hablar con el staff para ser considerados. En general, estamos abiertos y alentamos otros estudiantes a unirse a nosotros!

## 4 Clases y laboratorios

Las clases serán en el aula IS 2-2, en sistemas los días lunes de 2:30pm a 4:00pm y los viernes de y de 1:00pm a 2:30pm, respectivamente. Ustedes son responsables de todo el material impartido en las clases, incluyendo comentarios orales del profesor. Los labs serán impartidos por los ayudantes.

Los estudiantes también deben participar de 2 horas de laboratorio por semana, y también serán responsables por el material presentado en estos labs. Si van a los laboratorios, hay una gran posibilidad de que aprueben los certámenes. Muchos de los ejercicios que damos en los labs tienden a aparecer en los certámenes. Además, los labs les da la oportunidad de hacer preguntas e interactuar con el staff del curso de una manera más personalizada que durante la clase o por email. Los instructores del laboratorio les asignarán una nota que cuenta hacia la nota final.

Los labs serán los días miércoles y jueves a las 8:00am a 10:00am, en el Laboratorio de Ingeniería de Software, 3er piso de Sistemas.

Las clases serán en castellano, pero todo el material será en inglés, incluyendo mi interacción con la pizarra. Es importante que se acostumbren al inglés, ya que el 90% de la industria se mueve con este idioma como lengua franca. Sin embargo, ustedes no tienen que “producir” en inglés, sólo entender el inglés escrito. En el campus hay muchas opciones para aprender el idioma, incluyendo cursos del CFRD, otros alumnos el centro TeachIng, etc.

## 5 Tareas o “Assignments”

Se asignarán 3 problemas durante el semestre. Los problemas se sacan del juez online de la Universidad de Valladolid. El website del juez de la UVA es

[http://uva.onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8](http://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8)

Se sugiere a los estudiantes que visiten el sitio de la UVA, y que, si el tiempo les permite, empiecen a hacer algunos de los problemas que aparecen ahí. Esto les va a permitir practicar antes de submitir el ejercicio de LP1.

El clandario del curso muestra las fechas tentativas de la publicación de las tareas y sus fechas límite de entrega. La fecha de entrega igual estará confirmada en el email que les llegará con la publicación del problema en INFOALUMNOS. La tarea debe entregarse a las 11:59pm del día límite (usualmente 7 días después de la publicación de la misma).

El estudiante deberá submitir 3 archivos:

- Un archivo con extensión `.c` que contenga el código final del programa. El nombre de este archivo se dará con la publicación de la tarea. Si, por ejemplo, la tarea se llama `morse`, el archivo se llamará `morse.c`. Si trabajaron en equipo, deben colocar el o los nombres de los integrantes del equipo en el documento  $\text{\LaTeX}$ . Si trabajaron solos, sólo escriban “Miembros del equipo: Ninguno”.
- Un archivo  $\text{\LaTeX}$  que contenga la documentación del código, y una explicación de como llegaron a su solución. Si bien pueden juntarse a hacer el código en equipo, el documento de explicación es individual. Ver más abajo en la Sección 6.
- Un archivo `Makefile` para generar el código y la documentación en `pdf`. En la web del curso pueden encontrar un ejemplo de archivo `Makefile` con comentarios en su interior.

Todas las tareas se entregan a través de INFOALUMNOS. Los archivos de más arriba se entregan compactados en un solo archivo `.tar.gz`. Este archivo tiene que tener como nombre su número de estudiante y el sufijo  $Ax$ , donde  $x$  es el número de la tarea (1, 2, ó 3). Por ejemplo, `202012344A1.tar.gz` se refiere al estudiante con matrícula 202012344, quien acaba de submitir la tarea 1.

Luego de correr el programa `Makefile`, se generarán dos archivos, uno `pdf` y uno `.exe`, que es el programa en C compilado en un ejecutable (vea el `Makefile` para más instrucciones).

**IMPORTANTE:** Para aprobar el curso, la nota de las tareas deben tener promedio mayor o igual a 4. O sea, formalmente,  $(A_1 + A_2 + A_3)/3 \geq 4$ . Si todo el resto del curso fue aprobado, pero las tareas no, entonces quedan con la nota más alta, pero menor a 4 que tengan considerando todas las otras notas del curso. No hay reclamos por esto. **Si no tienen promedio mayor o igual a 4 en las tareas, no pasan el curso.**

La pauta de corrección de las tareas es la siguiente: se toma el programa, y se prueba con tres ejemplos. Uno de estos archivos de prueba es el ejemplo del problema en la web. Todo el mundo debería tener este ejemplo bien! (Si no, significa o que el autor no hizo realmente el problema y lo copió de un amigo, o que no lo compiló ni una vez, etc.). Los otros dos archivos de prueba son nuevos.

La siguiente itemización especifica la pauta de corrección con los valores asociados a cada ítem. Cada uno de estos valores *se descuenta* del 7 ideal.

- [ g ] General
  - [ g1 ] El trabajo no se entregó: -6pts.
  - [ g2 ] El trabajo se entregó después del deadline: -1pt, y -1pt por cada día adicional.
  - [ g3 ] Nombre/tipo de archivo incorrecto: -0.5pt
- [ s ] Sistema
  - [ s1 ] No compila: -4pts.
  - [ s2 ] Casos de prueba:
    - \* [ s2.1 ] El caso 1 no funciona (el ejemplo): -1pt
    - \* [ s2.2 ] El caso 2 no funciona (el ejemplo): -1.5pt
    - \* [ s2.3 ] El caso 3 no funciona (el ejemplo): -1.5pt
  - [ s4 ] Reportes de warning por el `-Wall`: -0.25pts por warning.
- [ d ] Documentación
  - [ d1 ] Error de  $\text{\LaTeX}$ : -2pts
  - [ d2 ] La explicación en si: esta es la única parte subjetiva de la evaluación. Se descontarán puntos por no claridad de comentarios, por código rebuscado, por cualquier cosa que nos sugiera a mí o a los ayudantes que hay algo que no entendieron, o que no pueden explicar lo que está pasando en el código. Sin embargo, si alguien ha hecho un muy buen trabajo, daremos puntos extras, aún si se pasa del 7. A veces, estos puntos definen si pasan o no la evaluación de las tareas, así que hagan un buen trabajo, vale la pena!

Algunas cosas que notar además:

- No existe una “recuperación” de las tareas. Las tareas son 3 y sólo 3.

- Después de submitir la tarea por INFODA, no se puede cambiar absolutamente nada después del día y hora límites. Aunque se equivoquen en una coma, si cuando los ayudantes o yo compilamos el programa, este no genera un ejecutable, o no se comporta como se espera (“but it works on my computer!”), ya es tarde para arreglarlo. Sin embargo, antes del cierre pueden submitir varias versiones si se dan cuenta de que se equivocaron.

## 6 Guía de redacción del informe de las tareas

Deben ser lo más claro posible en la documentación y explicación del código que van a entregar. De hecho, ser capaz de comunicar asuntos técnicos de manera efectiva es una habilidad muy importante que todo ingeniero debe tener en el set de herramientas.

Un programa simple vale más puntos que uno difícil de seguir y poco trabajado, además de que seguramente tiene menos errores. Trabajos que están desordenados recibirán menos puntos, aunque el programa esté correcto.

El informe se hará en  $\text{\LaTeX}$ . En la página web del curso se encuentra un modelo que se puede seguir. (De hecho, es muy parecido al de los escribas.)

Recuerden que el objetivo principal de este ejercicio es la comunicación. Se ha informado a los ayudantes que deben descontar puntos por documentos/programas que sean difíciles de seguir y exageradamente complejos.

## 7 Los certámenes

Este curso tiene 3 certámenes: dos certámenes teóricos y uno práctico en el laboratorio. Los certámenes teóricos suman el 50% de la nota final del curso. Sin embargo, tienen pesos diferentes. El segundo certamen es necesariamente (por acumulación y complejidad de contenidos) más difícil que el primero. Entonces, de esos 50% totales, el certamen 1 suma 40% (y dura 1.5 horas) y el certamen 2 el 60% (dura 3 horas). El certamen de laboratorio (certamen práctico) suma 15% directo a la nota final del curso, dura 1.5 horas y se hace enfrente de un computador en algún laboratorio.

Las fechas, horarios y lugares de los certámenes son tentativamente los siguientes:

Certamen 1	11 de mayo
Certamen 2	1 de julio
Certamen Lab	18 de junio
Recuperativo	6 de julio

Algunas cosas que notar además:

- Mis certámenes son realmente difíciles. Por favor, estudien acordemente.
- Para pasar el curso, los estudiantes tienen que tener un promedio  $\geq 4$  entre los dos certámenes *teóricos*.
- Hay un solo certamen recuperativo. Este certamen tendrá lugar durante la semana de recuperativos asignada por la UdeC: en este 2015, las evaluaciones de recuperación serán entre el 6 y el 15 de julio.

En la página web se encuentran certámenes de los años anteriores.

## 8 Los escribas

Todos los estudiantes deben transcribir una clase a partir de las notas de clase al papel (en español). Las notas deberán estar listas en el día de la clase siguiente. Vamos a pasar una lista para inscribirse durante la primera o segunda semana de clases.

Las notas deben estar escritas en L<sup>A</sup>T<sub>E</sub>X, usando el template en la web: <http://www.inf.udec.cl/~leo/lec-template.tex>.

Las notas de los escribas se envían vía email a la dirección del staff del curso como un archivo pdf, el original en L<sup>A</sup>T<sub>E</sub>X, y todas las figuras que se incluyeron en un archivo tar.gz. El archivo .tex debe llamarse `scribe_2015_1_503208_lectureXX.tex`, así como el tar.gz. La fecha límite para entregar las notas de una clase es al tercer día de esa semana. Es decir, si la clase fue un lunes, las notas se esperan el jueves de esa semana.

Seguramente el staff mandará comentarios, y posiblemente pidamos revisiones.

## 9 Pauta de nota final

Las notas se publicarán en INFOALUMNOS a medida que las vayamos teniendo. (**Consideren que corregir tareas y certámenes no es trivial, y demora un cierto tiempo. Así y todo, el staff hará lo posible para tener las notas lo antes posible**).

La nota final se obtendrá de las notas parciales de los certámenes teóricos y el práctico, de las tareas, y de la participación en clases (a partir de la participación como escriba, y en los foros de Piazza).

En particular:

Certamen 1 y 2	50% (40% y 60%)
Certamen Lab	15%
Tareas	30%
Participación	5% (incluye Piazza y escriba)

## 10 Políticas de colaboración

El objetivo de las tareas es que practiquen el material del curso. Entonces, se incentiva a que colaboren y hagan las tareas en grupo. Si trabajan en grupo, sin embargo, tienen el deber consigo mismo y con los otros miembros del grupo de estar preparados para la reunión del grupo. En principio, antes de dicha reunión, deberían pasar una hora tratando de resolver el problema solos. Si el grupo no puede resolver el problema, pueden hablar con otros grupos, o con los ayudantes, o postear dudas específicas online.

Sin embargo, **el documento de explicación del código deben escribirlo solos, sin ninguna clase de ayuda de parte de nadie más**. Aún si colaboraron con alguien más. Si se les pide que identifiquen con quién trabajaron en grupo, si lo hizo sólo, solamente coloque “colaboradores: ninguno”. Si tomaron algo de la web, por favor también den crédito a la fuente, de la mejor manera escolástica. Así y todo, tienen que escribir la solución de la tarea en sus propias palabras. **Ustedes deben ser capaces de explicar el código con sus propias palabras al staff del curso.** El plagio y otra conducta deshonestas no será tolerada en la universidad. Si tienen preguntas sobre la política de colaboración del curso, por favor pregunten por Piazza. Si tienen dudas de haber infringido en la política de colaboración, por favor acérquense a hablar con

el staff. Si bien los casos de plagio y copia deben ser manejados severamente, usualmente somos más comprensivos si ustedes nos cuentan que si nos enteramos por terceros.

**Los certámenes teóricos y prácticos son individuales.** No se permite la colaboración en ellos.

## 11 Libro

La referencia por excelencia de este curso es:

Brian W. Kernighan and Dennis Ritchie. 1988. *The C Programming Language* (2nd ed.). Prentice Hall Professional Technical Reference.

La biblioteca tiene varias copias del libro en español, y hay varias versiones online. Para revisar, daremos las correspondientes secciones (p. ej. 1.2 de K&R), en vez de páginas, porque las páginas pueden variar con respecto a las ediciones con las que estén estudiando.

Así y todo, lo que más vale son las notas del curso, que pondremos en la web a medida que avance el curso. Hay mucho más material en estas slides, contando ejemplos que el staff estima necesario para que entiendan como un computador hace lo que le pedimos que haga a través de un programa. Entonces, el libro es la referencia básica, pero las notas y las slides son imprescindibles.

**Cualquier cosa que se diga en la clase, o cubra en el libro o las notas o el lab puede ser evaluado.**

## 12 Software y recursos web

Las tecnologías de este curso son las siguientes: Linux (sugiero Ubuntu), gcc, L<sup>A</sup>T<sub>E</sub>Xy algún editor de texto, yo sugiero Emacs. Una vez instalado Linux, a fin de tener todo lo necesario se debe hacer

- `sudo apt-get install emacs`
- `sudo apt-get install texlive`
- `sudo apt-get install binutils build-essential manpages-posix-dev manpages-dev`

La página web del curso ([www.leoferres.info/2015\\_1\\_503208.html](http://www.leoferres.info/2015_1_503208.html)) contiene links a versiones electrónicas de todos los handouts, las correcciones al material y notas del curso, y anuncios especiales. Háganse un tiempo todas las semanas para visitar los diferentes sitios asociados al curso: INFOALUMNOS (para notas y publicación de tareas, etc.), Piazza (para el foro, anuncios, y para la participación en clase), la página web propia tiene también recursos como cheatsheets, software y otros links de interés.

## 13 Filmación del curso

En el 2015, el staff tiene como objetivo filmar nuevamente el curso, como se hizo hace dos años, durante el primer semestre de 2013. Ahora tenemos mejor equipamiento y hemos aprendido de nuestros errores.

La cámara esta usualmente apuntando al instructor del curso, y los alumnos no debieran aparecer en la filmación. Sin embargo, si alguno tiene problemas en aparecer en la filmación, por favor siéntense atrás en la sala.

Los videos deberían aparecer durante la misma semana de la filmación en un link a ser difundido por la página web del curso, Piazza y el mail de INFOALUMNOS.

## 14 Sobre el método de enseñanza

El uso extensivo del pizarrón es una decisión consciente:

- Les permite a los estudiantes pensar preguntas mientras se escribe
- Les permite a los estudiantes tener tiempo para tomar notas
- Les permite a los estudiantes saber qué es crítico de lo que se está diciendo en clase
- Les permite a los estudiantes saber que el instructor también tiene que “trabajar” en escribir
- Les permite a los estudiantes procesar lo que se dice, en mi opinión, mucho más que con Powerpoint.
- Les permite a los estudiantes “seguir” al profesor sin tener que mirar las slides/diapos

Además, trabajar con pizarra tiene un tinte “old school” que entretiene, en nuestra opinión, y que hace más fácil luego seguir los videos por la web.

Este año también escribiremos los problems y el código en las pizarras, en vez del computador, como si fueran fórmulas matemáticas. Toda la práctica se dará en los laboratorios, y los códigos demostrados en los labs y en clases estarán disponibles en `github` en <https://github.com/leoferres/LP1>. Sin embargo, se espera que el estudiante transcriba o al menos compile y ejecute estos códigos de ejemplo porque muchos contienen varias lecciones que vale la pena aprender.

## 15 Ayuda extra

Esperamos que los estudiantes publiquen sus preguntas (y ayuden a contestar las preguntas de sus compañeros) en Piazza en [piazza.com/udec.c1/spring2015/503208/home](https://piazza.com/udec.c1/spring2015/503208/home).

También el staff pondrá horarios de oficina para que los alumnos puedan ir a hacer sus preguntas más como 1:1. Pueden ir a cualquier horario de cualquiera de los miembros del staff.

Si hay suficiente interés, Este año por primera vez haremos un Linux Install Fest, por si alguno de ustedes quiere traer su laptop o desktop a instalar Linux (Ubuntu) con nosotros un sábado a la mañana. El Linux Fest se hará durante el primer o segundo fin de semana, en horario a definir y publicar en la web.

Durante el Linux Install Fest, haremos también un mini-curso de `gcc`, `emacs` y  $\text{\LaTeX}$ . El objetivo es producir el programa `hello.c`, con el tradicional ejemplo “Hello, World!”, y un “hello world” en `pdf` usando  $\text{\LaTeX}$ .

**Este curso tiene material genial  
Hagamos que sea el mejor curso de  
programación de la UdeC!!!**