Tema 0: Presentación de la asignatura

Diapositivas de clase.

- Aquí puedes descargar las diapositivas utilizadas en clase, que prácticamente contienen lo mismo que la web: diapositivas tema 0

Profesorado.

Teoría.

- Francisco J. Castellanos Regalado. DLSI.

Prácticas.

- Antonio-M Corbí Bellot, Alejandro Reina Reina. DLSI.

Tutorías.

- Debes solicitar tutoría a través de UA-Cloud, ya sea presencial o virtual.
- Para dudas de teoría, acude al profesor de teoría, mientras que para dudas de prácticas, consulta a tu profesor de prácticas.

Contexto.

- Primer curso: Programación I , primer semestre
- Primer curso: Programación II , segundo semestre.

> La asignatura dispone de página web propia. **Es recomendable que la consultes habitualmente**.

- La asignatura consta de dos bloques:
  - Teoría.
  - Prácticas.
- Cada uno de ellos tiene un peso del 50% en la nota final.

Primer periodo de evaluación (C3).

- Para promediar teoría y prácticas es necesario obtener como mínimo un 4.0 en ambos bloques.
- Para superar la asignatura, la nota final debe ser igual o superior a 5.0.
- El *bloque de teoría* consta de un único examen final.
- Es un examen tipo test del tipo *elegir la única respuesta posible entre cuatro alternativas*.

  Tres respuestas erróneas anulan una correcta.

  Es conveniente hacerlo a lápiz ya que puedes corregir respuestas previas que hayas marcado y quieras cambiar.
- El *bloque de prácticas* consta de:

  1. Evaluación continua (4 prácticas) con un peso del 20% sobre la nota final.
  2. Un examen práctico final realizado en el laboratorio y con un peso del 30% sobre la nota final. Los ejercicios de los que constará serán similares a los realizados en las prácticas hechas durante el curso.
- La **asistencia** a las sesiones **prácticas** es **obligatoria**. Se permite un máximo de 4 faltas justificadas.

Segundo período de evaluación (C4).

- Si no se supera el primer período de evaluación entonces en la *convocatoria C4*:

bloque en el que tenga una 4.0 <= nota < 5.0 en la *convocatoria C3*.

La decisión de presentarse a dicho examen corresponde al propio alumno teniendo en cuenta que la nota obtenida en esa prueba en esta convocatoria sustituye a la obtenida en la C3.

- Para promediar teoría y prácticas es necesario obtener como mínimo un 4.0 en ambos bloques.

- El examen de prácticas será un examen práctico realizado en el laboratorio, su contenido será del mismo estilo que en la *convocatoria de junio*. **En esta convocatoria la nota de este examen será la nota total del bloque de prácticas**, es decir, el 50% de la nota final.

Sistema de evaluación que se aplicará en la convocatoria de diciembre.

- Se realizará un examen de teoría y uno de prácticas, ambos similares a los de la *convocatoria de julio*.
- Para promediar teoría y prácticas es necesario obtener como mínimo un 4.0 en ambos bloques.

Importante.

1. En cualquier apartado puntuable de la asignatura una calificación de **3.99 no es un 4.0**.

2. No se admiten entregas de prácticas fuera de plazo o por otro cauce que no sea el indicado en el enunciado. Si esto ocurriera esa entrega será ignorada, tenlo en cuenta.

3. La publicación de notas se realiza en la web del DLSI.

4. Si necesitas revisar cualquier práctica de la asignatura debes reservar una tutoría presencial desde la aplicación de UA Cloud.

5. La corrección de las prácticas se realiza con la versión del compilador de `c++` que tenéis instalada en los laboratorios de la EPS.

hacer que tu código no compile o funcione de manera diferente a como lo haría con la versión del compilador de la EPS.

7. En la asignatura empleamos la versión de C++ de 2017, más conocida como C++17 y más concretamente la que implementa el compilador instalado en los laboratorios de la EPS.

   Una forma sencilla de tenerlo instalado es usando una *máquina virtual* de VirtualBox. Para ello, debes crear una imagen [[https://en.wikipedia.org/wiki/VirtualBox#VirtualBox%5FDisk%5FImage][VDI]] de la instalación de Ubuntu en la EPS (Ubuntu 24.04). Hay mucha documentación referida a la creación de máquinas virtuales, por ejemplo [[https://osl.ugr.es/2020/09/29/como-instalar-ubuntu-en-virtual-box/][aquí]].

8. En la asignatura se realiza un *control de copias* con cada entrega de prácticas.

9. Una práctica copiada supondrá tener una nota de **cero** en la misma para todos los alumnos implicados. Además, se informará a las direcciones del DLSI y de la EPS de ello para que se tomen las medidas disciplinarias oportunas.

10. Es conveniente que leas el Reglamento para la evaluación de los aprendizajes de la UA, p.e en el artículo 14 donde se habla de *Realización fraudulenta de pruebas de evaluación*, punto 1:

    > Durante la prueba, el alumnado está obligado a observar las normas sobre autenticidad del ejercicio y privacidad del mismo.

11. Fíjate que:

    1. Copiar una práctica de otra persona total o parcialmente incumple el 14.1 (autenticidad).
    2. Difundir una práctica (para que otros se la copien o se inspiren), también incumple el 14.1 (privacidad)

Teoría.

datos.

- El lenguaje de programación **C++** (**C++98**, **C++11**, **C++14**, **C++17**, **C++20** y **C++23**).
- Gestión y uso de memoria basada en la pila (stack), en almacenamiento global y en almacenamiento dinámico.
- Introducción a los Tipos Abstractos de Datos (*TAD*).

TEMA 2: Tipos abstractos de datos: Listas (simples, dobles), Pilas, Colas.

- Definición.
- Operaciones básicas.
- Diferentes implementaciones.

TEMA 3: Tipos abstractos de datos: Árboles, Grafos.

- Definición.
- Operaciones básicas.
- Diferentes implementaciones.
- Tipos de árboles. Tipos de grafos.

TEMA 4: El paradigma orientado a objetos. Características básicas de los lenguajes orientados a objetos.

- Presentación y justificación del modelo de programación orientada a objetos.
- Requisitos para que un lenguaje se considere orientado a objetos.
- **C++** como lenguaje orientado a objetos.

TEMA 5: Clases y objetos. Espacios de nombres.

- Definición del concepto de *clase* y del concepto de *objeto*.
- Diferentes tipos de clases (*abstractas*, *metaclases*, *interfaces*, etc...).
- Concepto de *espacio de nombres*. Agrupación de símbolos en espacios de nombres.

- ¿Qué es un evento? ¿Cómo se producen?
- Cómo cambia nuestro estilo de programación.
- Cómo realizarla tanto desde **C** como desde **C++**.

TEMA 7: Relaciones entre objetos. Herencia. Polimorfismo. Enlace dinámico.

- Tipos de relaciones entre objetos en ausencia de herencia (uso, composición).
- Concepto de *herencia* entre clases.
- Herencia de implementación y herencia de interfaz.
- Relación **Es un** entre objetos. Concepto de polimorfismo.
- Tipos de herencia (simple, múltiple).
- Implicaciones de la herencia entre clases y la relación **Es un** entre objetos.
- Qué es el *enlace dinámico*. Ventajas e inconvenientes.
- Cómo influye en el diseño de lenguajes orientados a objetos (virtual, final).

TEMA 8: Genericidad.

- Concepto de Genericidad. Características y usos.
- La genericidad como otro tipo de polimorfismo.
- Metaprogramación: ejecución de código en tiempo de compilación.

TEMA 9: Excepciones. Patrón *RAII*.

- Tratamiento de errores bajo el paradigma orientado a objetos.
- Qué es una excepción, Jerarquía de clases de excepciones.
- Cómo influye en el diseño de lenguajes orientados a objetos (`try` , `throw` , `catch` , `finally`).

orientados a objetos.

- Cómo implementan diferentes lenguajes de programación las características explicadas hasta ahora. **Java**, **C#**, **D**, **Python**, etc...

TEMA 11: POO con lenguajes no orientados a objetos.

- Usando sólo el lenguaje **C** simularemos la herencia de clases, paso de mensajes, enlace dinámico y variables de clase.

TEMA 12: Pruebas unitarias.

- Aprenderemos a pasar tests de forma sistemática a nuestro código. Automatización con el uso de CTest (CMake).

Prácticas.

- En la parte de evaluación continua tenemos *cuatro prácticas individuales*:
  - **Práctica 1:** Ejercicios de repaso de Programación-I.
  - **Práctica 2:** Implementación y uso de TADs.
  - **Práctica 3:** Programación Dirigida por Eventos.
  - **Práctica 4:** Programación Orientada a Objetos.

Disponemos también de una **Práctica 0** la cual nos sirve como práctica de iniciación en la asignatura.

No cuenta para la nota final.

- Además de un examen individual de prácticas en el laboratorio. Este examen se hace el mismo día del examen de teoría.

- A partir del curso 2022/2023 la asignatura va a proporcionaros *docencia complementaria* al inicio de algunas clases de prácticas.

valgrind.

Estas tres herramientas ya las veníamos explicando desde hace algunos cursos.

2. Trabajo con el S.O. desde un terminal: Bash shell script
3. Doxygen para generar la documentación de tu código a partir de ciertos comentarios puestos en él.
4. Git para realizar control de versiones. Solo veremos el uso local del mismo.
5. Dear ImGui para crear aplicaciones con *Interfaz Gráfico de Usuario*. Esta parte depende de que hayamos visto lo correspondiente a Programación Orientada a Objetos.

Entrega de prácticas.

- Se realizará en https://pracdlsi.dlsi.ua.es y en el plazo de tiempo allí establecido.
- Se pueden realizar tantas entregas como se quiera, sólo se corregirá la última.
- Es necesario seguir estrictamente las instrucciones de los enunciados de las prácticas (especialmente en lo referente al formato de la salida).
- Ten en cuenta lo anterior pues **No se admitirán entregas por otro *conducto* y/o fuera de plazo**.

Corrección de las prácticas.

- **Será automática**. La *no compilación* o *no enlace* del código implicará una nota de 0.0, los errores en tiempo de ejecución que causen una finalización anómala de la práctica (`violación de segmento, división por cero, etc...`) implicarán una nota de 0.0.
- Para revisar la corrección de tus prácticas reserva una tutoría presencial desde la app de UA-Cloud con tu profesor de prácticas.
- **No se debe entregar** un programa principal. Ya lo proporcionará el programa corrector correspondiente de la práctica.
- **Importante:** Debeis respetar:

- Uso de mayúsculas/minúsculas en nombres de archivos.

- **Recordad**, *no es lo mismo* que una función muestre un resultado por pantalla (`printf` , `cout`) a que lo devuelva.

Planificación semanal.

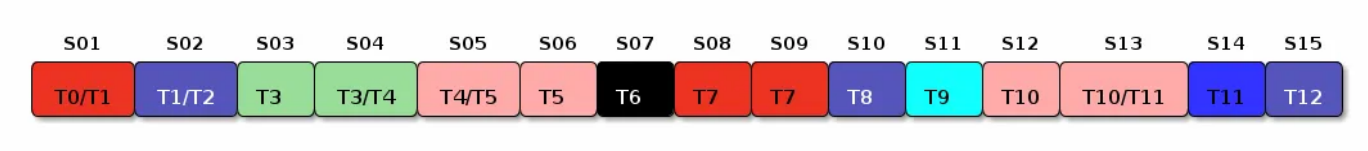

Figura 1: Planificación de sesiones de teoría.

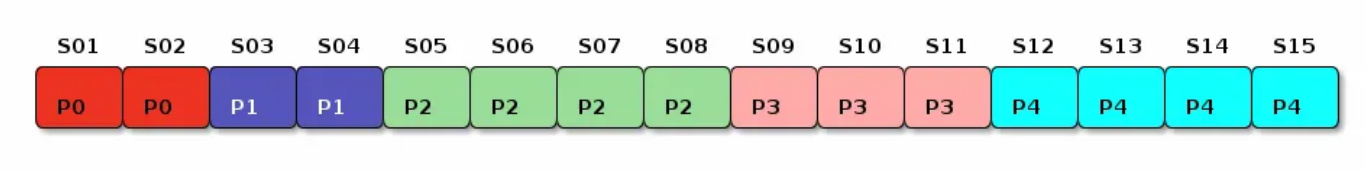

Figura 2: Planificación de sesiones de prácticas.

Consejos.

- La asignatura no es difícil pero sí tiene muchos conceptos nuevos; *estudia día a día*.

  Para que te sirva como referencia... **por cada hora de clase** deberías dedicarle alrededor de **hora y media de estudio en casa**.

- Aprovecha las tutorías, realiza tantas como necesites.

- Dispondréis con suficiente antelación de las transparencias empleadas en clase de teoría para facilitaros el seguimiento de las mismas.

P2 GIR

En esta página > Profesorado.