|  | **Carrera:** Técnico Universitario en Programación  **Materia:** Programación II  **Tema:** Contrato Pedagógico |
| --- | --- |

**Contrato Pedagógico - Programación II**

**Integrantes de la cátedra**

Kloster, Daniel - Profesor

Wenner, Maximiliano - Ayudante

**Características**

La materia está organizada en 16 encuentros que se desarrollarán de manera presencial y/o por videoconferencia en la plataforma Zoom los siguientes días y horarios:

* **Lunes de 18:00 a 22:00.**
* **Miércoles de 08:00 a 12:00.**

Aunque se lo recomienda, la participación en los encuentros virtuales no es obligatoria, ya que serán grabados y estarán a disposición de los participantes para que puedan ser vistos en cualquier horario. En los encuentros presenciales también se grabarán las clases si la disponibilidad de elementos técnicos lo permite.

En los encuentros se podrán abordar contenidos teóricos y/o prácticos. Más abajo se hace una descripción de los temas a tratar en cada una de las clases. De acuerdo a cómo se desarrolle el curso, los temas y actividades podrán sufrir alguna modificación.

Toda la información del trabajo en la materia estará accesible en el Aula Virtual. Se espera establecer a partir de ella una interacción entre docentes y estudiantes. Por las características especiales propias de la modalidad de cursada no presencial, se propone que el Aula Virtual se convierta en el espacio central de información, consulta, encuentro y participación entre todos los participantes de la materia.

**Contenido**

La materia se apoya en los contenidos vistos en Programación I y Laboratorio I, y aborda temáticas más avanzadas desde la perspectiva de la programación estructurada, para pasar luego a la Programación Orientada a Objetos.

El contenido completo puede verse en el [**Programa de la materia**](https://docs.google.com/document/d/1HIyf8f-5pr3Pc5kCEXyIm5PWK4BYrE6IWHqc8Q5BiXg/edit?usp=sharing)

**Evaluación**

**Parciales**

Se tomarán dos exámenes parciales, que integrarán los contenidos vistos hasta ese momento en el curso.

Ambos parciales consisten en la resolución de un conjunto de enunciados problema similares a los resueltos en los encuentros previos. El tiempo máximo para el desarrollo de los exámenes será de dos horas.

Al examen se accederá desde el Aula Virtual en una fecha previamente establecida, y en el horario de los encuentros. Cada alumno deberá resolver las preguntas contenidas en el cuestionario del parcial, y luego subir las soluciones como se indique en las consignas.

La participación en los parciales es obligatoria, así como la permanencia en el Zoom durante todo el tiempo de desarrollo de los exámenes.

Los parciales se califican con notas entre 1 y 10, siendo el 6 el valor mínimo para aprobar correspondiendo al 60% del examen satisfactoriamente desarrollado.

Las notas definitivas de las evaluaciones se publicarán una semana después de tomar el examen, aunque durante el encuentro en el que se desarrollarán cada alumno/a tendrá una nota aproximada de su desempeño. La nota podrá obtenerse de manera automática si la evaluación se realiza mediante un cuestionario preparado para eso, o mediante un proceso de autoevaluación que formará parte de la evaluación misma. La calificación podrá visualizarse en el Aula Virtual y luego en el sistema SysAcad.

En la clase posterior al examen se procederá a realizar un análisis de los ejercicios propuestos y su corrección. **En esa clase quien así lo considere necesario podrá solicitar la revisión individual o hacer consultas sobre el parcial. Fuera de esa ocasión, sólo podrán pedir revisión quienes no hayan podido participar por problemas debidamente justificados. NO SE RESPONDERÁN MENSAJES CON CONSULTAS SOBRE LOS RESULTADOS.**

**Recuperatorio**

La materia contará un recuperatorio que integra todos los temas vistos. Tendrán similares características a los parciales.

**Criterios de regularización y promoción**

Para **regularizar la materia** se deberá obtener una **calificación de 6 o más en ambos parciales** o haber aprobado el recuperatorio.

Para **promocionar la materia** se deberá obtener una **calificación promedio de 8 ó más entre los parciales/recuperatorio aprobado/s.**

Si se accede a la instancia de recuperatorio, se podrá promocionar siempre y cuando la nota del recuperatorio sea 8 o más y que se haya aprobado en la primera instancia alguno de los parciales.

Si no se aprueba ninguno de los parciales, igual se puede dar el recuperatorio. En este caso, si se aprueba el recuperatorio se podrá regularizar la materia, pero no promocionarla.

En todos los casos la nota de la promoción surge del promedio de notas entre todos los parciales.

**Examen Final**

El examen final de la materia es la evaluación de todos los contenidos vistos durante el curso. Para acceder a esta instancia deberán haber cumplido con los criterios de regularización. Luego, será necesario inscribirse a una mesa de final a través del sistema SysAcad.

El examen final de Programación II consiste en el desarrollo de un examen similar al resuelto en el segundo parcial. La evaluación es teórico-práctica.

**Nota importante:** tanto el final como la defensa del trabajo es individual, y en el caso de que éste sea mediante Zoom, debe hacerse con el micrófono y la cámara del equipo habilitada. Cualquier inconveniente técnico o de cualquier otra índole sobre el punto anterior debe plantearse a los docentes con anterioridad. SIEMPRE ES POSIBLE DAR EL EXAMEN EN LA SEDE DE LA FACULTAD

**Contenidos de las clases**

Los temas a tratar en las clases se dispondrán aproximadamente de la siguiente manera:

**Curso lunes**

| Clase | Fecha | Temas |
| --- | --- | --- |
| 1 | 18/03 | • Tipo de datos primitivos de C/C++.  • Definición de variables. Alcance local y global  • Vectores.  • Punteros. Relación entre vectores y punteros  • Matrices: definición. Uso  • Matrices bidimensionales y multidimensionales.  • Cadenas de caracteres. Funciones básicas para cadenas.  • Funciones: prototipos de funciones, semántica. Diseño de funciones, paso de parámetros por valor y dirección |
| 2 | 25/03 | • Funciones: sobrecarga de funciones. Paso de parámetros por referencia. Valores por omisión para los parámetros.  • Ejercicios con vectores, matrices y cadenas de caracteres  • Tipo de dato estructurado registro (struct):  • Definición y declaración. Uso. Acceso a los campos  • Introducción a POO:  • Diferencias entre Prog. Est. y P.O.O  • Definición de clases  • Definición de objetos  • Implementación. Definición de propiedades y métodos.  • Especificadores de acceso  • Conceptos de encapsulamiento, abstracción y polimorfismo. |
| 3 | 01/04 | FERIADO |
| 4 | 08/04 | Clases y objetos en C++:  • Constructores y destructores  • Diseño de clases  • Criterios básicos para establecer propiedades y métodos.  • Puntero this  • Mecanismos útiles para el diseño de clases: composición y herencia |
| 5 | 15/04 | Archivos:  • Definición y uso de archivos.  • Funciones básicas:  • fopen(), fwrite(), fread(), fclose()  • Operador sizeof.  • Estructura FILE.  • Ejercicios con archivos |
| 6 | 22/04 | Archivos:  • Operaciones con archivos: alta, baja lógica. Listados.  • Ejercitación. |
| 7 | 29/04 | Archivos:  • Operaciones con archivos: concepto de baja física, modificación de campos.  • Funciones fseek() y ftell()  • Ejercitación. |
| 8 | 06/05 | • Repaso general. Ejercicios integradores |
| 9 | 13/05 | **Primer parcial** |
| 10 | 20/05 | Entrega y análisis del primer parcial.  • Asignación dinámica de memoria: operadores new y delete  • Clases que generan objetos de manera dinámica. Uso de constructores y destructores para crear y eliminar objetos.  Clase Cadena.  Ejercitación |
| 11 | 27/05 | Clases:  Sobrecarga de operadores y funciones.  • Funciones amigas.  • Uso avanzado del mecanismo de la herencia: polimorfismo. |
| 12 | 03/06 | Ejercicios integradores  Modelo de parcial |
| 13 | 10/06 | **Segundo parcial.** |
| 14 | 17/06 | **FERIADO.** |
| 14 | 24/06 | Entrega y análisis segundo parcial  Entrega de TP integrador |
| 15 | 01/07 | **Recuperatorio** |
|  | 08/07 | **RECESO** |
| Final 1 | 15/07 | **Entrega resultado recuperatorio**  **Final.** |
| Final 2 | 22/07 | **Final** |

**Curso miércoles**

| Clase | Fecha | Temas |
| --- | --- | --- |
| 1 | 20/03 | • Tipo de datos primitivos de C.  • Definición de variables. Alcance local y global  • Vectores.  • Punteros. Relación entre vectores y punteros  • Matrices: definición. Uso  • Matrices bidimensionales y multidimensionales.  • Cadenas de caracteres. Funciones básicas para cadenas.  • Funciones: prototipos de funciones, semántica. Diseño de funciones, paso de parámetros por valor y dirección |
| 2 | 27/03 | • Funciones: sobrecarga de funciones. Paso de parámetros por referencia. Valores por omisión para los parámetros.  • Ejercicios con vectores, matrices y cadenas de caracteres  • Tipo de dato estructurado registro (struct):  • Definición y declaración. Uso. Acceso a los campos  • Diferencias entre Prog. Est. y P.O.O  • Introducción a POO:  • Definición de clases  • Definición de objetos  • Implementación. Definición de propiedades y métodos.  • Especificadores de acceso  • Conceptos de abstracción y polimorfismo.  • Constructores y destructores |
| 3 | 03/04 | Clases y objetos en C++:  • Puntero this  • Diseño de clases. Criterios básicos para establecer propiedades y métodos.  • Mecanismos útiles para el diseño de clases: composición y herencia |
| 4 | 10/04 | Archivos:  • Definición y uso de archivos.  • Funciones básicas:  • fopen(), fwrite(), fread(), fclose()  • Operador sizeof.  • Estructura FILE.  • Ejercicios con archivos |
| 5 | 17/04 | Archivos:  • Operaciones con archivos: alta, baja lógica. Listados.  • Ejercitación. |
| 6 | 24/04 | Archivos:  • Operaciones con archivos: concepto de baja física, modificación de campos.  • Funciones fseek() y ftell()  • Ejercitación. |
| 7 | 01/05 | FERIADO |
| 8 | 08/05 | • Repaso general. Ejercicios integradores |
| 9 | 15/05 | **Primer parcial.** |
| 10 | 22/05 | Entrega y consultas acerca del primer parcial.  Clases  • Asignación dinámica de memoria: operadores new y delete  • Clases que generan objetos de manera dinámica. Uso de constructores y destructores para crear y eliminar objetos.  Clase Cadena |
| 11 | 29/05 | Clases:  Sobrecarga de operadores y funciones.  • Funciones amigas.  • Uso avanzado del mecanismo de la herencia: polimorfismo.  Ejercitación |
| 12 | 05/06 | Clases:  Estructura de datos dinámicas:  • Listas simples y enlazadas.  • TDA Cola  • TDA Pila  Ejercitación |
| 13 | 12/06 | Ejercicios integradores. Repaso general |
| 14 | 19/06 | **Segundo parcial.** |
| 15 | 26/06 | Corrección parcial 2.  Entrega TP integrador |
| 16 | 03/07 | Recuperatorio integral |
|  | 10/07 | **Receso** |
| Final 1 | 17/07 | Resultado y corrección recuperatorio integrador  Final. |
| Final 2 | 24/07 | Final. |