

PROGRAMACIÓN ESTRUCTURADA

PLANEACIÓN DEL CURSO

TRIMESTRE 24-I ◊ 5/MARZO–31/MAYO

Información general

U.E.A. Programación Estructurada	PROFESOR	Asael Fabian Martínez Martínez
GRUPO CZ12 CLAVE 2137075 CRÉDITOS 9	E-MAIL	amartinez@izt.uam.mx
HORARIO 9:30–11:00 • lunes, miércoles, viernes		fabian@xanum.uam.mx
TOTAL HORAS 66	CUBÍCULO	AT321

Descripción del curso

Descripción general

En este curso se proporcionarán las bases para el diseño e implementación de programas que resuelvan problemas algorítmicos bajo el paradigma de la programación estructurada. Para ello se estudiarán diversos ejemplos de problemas que pueden ser resueltos computacionalmente, así como los principales algoritmos de ordenamiento y búsqueda, tanto iterativos como recursivos. Todos los conceptos estudiados se implementarán utilizando un lenguaje de programación de alto nivel (e.g. Python o C).

Contenido sintético

1. Principios de programación estructurada

Introducción a la computación y a las computadoras — procesamiento de información — algoritmos — pseudocódigo — lenguajes de programación.

2. Elementos de programación

Introducción — tipos de datos simples — estructuras de control secuencial, selectivas e iterativas — procedimientos, funciones y paso de parámetros — bibliotecas de funciones — tipos de datos estructurados — manejo de archivos — diseño ascendente y descendente — documentación.

3. Recursividad

Objetivo general

Plantear soluciones algorítmicas a problemas utilizando la programación estructurada.

Utilizar adecuadamente los tipos de datos simples, estructurados y abstractos en el diseño de algoritmos.

Aplicar correctamente el diseño modular en la implementación de algoritmos.

Familiarizarse con el uso de bibliotecas de funciones.

4. Algoritmos de ordenamiento y búsqueda

Algoritmos de ordenamiento por inserción directa y selección directa — búsqueda secuencial — búsqueda binaria — algoritmos recursivos de ordenamiento y de búsqueda.

5. Tipos de datos abstractos

Conceptos básicos — definición e implementación — tipos de datos abstractos lineales (listas, pilas, colas) — tipos de datos abstractos no lineales (árboles, grafos).

6. Bibliotecas de funciones

Aspectos y criterios de evaluación

El total de la calificación se dividirá como sigue:

Prácticas y tareas	25%
Proyectos parciales	25%
Exposición	10%
Proyecto final	30%
Asistencia	10%

Se tendrán prácticas semanales, tentativamente los días viernes, en donde se aplicarán los temas vistos a lo largo de la semana. Estos ejercicios tendrán que ser entregados para su evaluación.

Adicional a las tareas, se tendrán dos proyectos parciales, que consistirán en dar solución a problemas específicos. La demanda para resolverlos será mayor que la requerida para las prácticas y tareas.

La exposición será sobre algún algoritmo de ordenamiento o búsqueda, algún tipo abstracto de datos o alguna biblioteca de funciones.

En el proyecto final se aplicarán todos los conocimientos adquiridos durante el curso para dar solución a algún problema específico. Además de la entrega de los códigos correspondientes, se tendrá que realizar una presentación oral del problema y su solución.

La escala de calificación será de 0 a 10 y su equivalente en letra es la siguiente:

<6.0	⇒ NA
[6, 7.8)	⇒ S
[7.7, 8.9)	⇒ B
≥8.9	⇒ MB

Lenguaje de programación

Como se mencionó, todos los conceptos estudiados en clase se implementarán utilizando un lenguaje de programación de alto nivel, por ejemplo Python o C. Es altamente

recomendable que todas las actividades (prácticas, tareas, proyectos, ...) se entreguen en alguno de estos lenguajes; sin embargo, existe la libertad de trabajar con otro lenguaje imperativo u orientado a objetos. En el segundo caso, será necesario informar al profesor del lenguaje que se utilizará. Con relación a la entrega de actividades, se presentan algunas consideraciones mínimas que deben tenerse en cuenta.

1. Para tener derecho a una calificación aprobatoria, se deberán entregar todas las actividades, sin importar su calificación.
2. Cada tarea tendrá una fecha límite de entrega. No se aceptarán entregas posteriores.
3. Tareas iguales, muy parecidas o descargadas de internet, total o parcialmente, serán anuladas; se calificará con NA a quien tenga dos o más tareas anuladas.
4. Se cuenta con un aula virtual, alojada en Virtuami

<https://virtuami.izt.uam.mx/aulas/avmacca/>

en donde se podrán descargar algunos de los materiales para el curso y enviar tareas.

5. Considerar los posibles inconvenientes que puedan surgir (como la falta de internet o luz, o archivos demasiado grandes) para la entrega puntual de las tareas.

Sobre la entrega de programas:

6. Se deberán entregar dos archivos: *código fuente* y *resultados*.
7. El *código fuente* deberá estar propiamente documentado; incluir al inicio de cada archivo el nombre de quien entrega.
8. Los *resultados* consisten en la salida del programa y se deberán entregar en un archivo en formato pdf.

Bibliografía

1. Hinojosa, A. P. 2022, El lenguaje de programación Python de principio a fin.
2. Joyanes, L. y Zahonero, I., 2014, Programación en C, C++, Java y UML, Segunda edición. Mc Graw Hill.