IFE Educabol
—
WhatsApp: +591 75553338 · Web: ife.bo
Email:
Redes: Facebook: ite.educabol WhatsApp: 59175553338 Instagram: ite.educabol
WhatsApp: 59171039910 TikTok: @ite_educabol WhatsApp: 59171324941

Lunes a Viernes

Inversión	Bs 380.00
Descripción	Entrenamiento diario: ejercicios, retos y revisión de código.

Horarios disponibles

Horario	Estado
07:30 - 09:00	**SIN CUPO**
09:00 - 10:30	**SIN CUPO**
10:30 - 12:00	**Disponible**
14:00 - 15:30	**SIN CUPO**
15:30 - 17:00	**SIN CUPO**
17:00 - 18:30	**Disponible**

Días de asistencia

Día

Beneficios (generales)

Beneficio	Detalle
Pensamiento algorítmico	Resolver problemas paso a paso con estructuras y funciones.
Multilenguaje	Aplicar conceptos base en C++, Java, JavaScript y Python.
Estructuras y árboles	Dominar listas, pilas, colas y árboles (recorridos e inserción).

Materiales sugeridos

Material	Descripción
VS Code / IDEs	Editor con extensiones útiles y/o IDE por lenguaje.
Compiladores/Intérpretes	g++ (C++), JDK (Java), Node.js (JS), Python 3 (Python).

Contenidos (Nivel Inicial)

Subnivel	Título	Descripción
1. Entorno y Hola Mundo	1. Entorno y "Hola Mundo"	Panorama general del entorno y objetivo del primer programa.
1. Entorno y Hola Mundo	1.1 Instalación de herramientas (g++, JDK, Node.js, Python)	Instalar compiladores e intérpretes para C++, Java, JS y Python.
1. Entorno y Hola Mundo	1.2 Instalación y configuración de VS Code	Extensiones recomendadas, configuración básica y depuración inicial.
1. Entorno y Hola Mundo	1.3 Instalación y uso básico de Git	Git init, commit, status y sincronización simple.
1. Entorno y Hola Mundo	1.4 Compilar y ejecutar programas en los 4 lenguajes	Comandos de compilación/ejecución para C++, Java, JS y Python.
1. Entorno y Hola Mundo	1.5 Primer programa "Hola Mundo" en cada lenguaje	Imprimir en consola el clásico Hola Mundo en cada stack.
2. Secuencias y Entrada/Salida	2. Secuencias y Entrada/Salida	Introducción a flujo de ejecución y E/S estándar.
2. Secuencias y Entrada/Salida	2.1 Flujo de ejecución	Orden de instrucciones y efectos laterales.
2. Secuencias y Entrada/Salida	2.2 Operadores (aritméticos, relacionales, lógicos)	Precedencia y asociatividad; cortocircuito lógico.
2. Secuencias y Entrada/Salida	2.3 Entrada y salida estándar	Lectura/escritura en consola por lenguaje.
2. Secuencias y Entrada/Salida	2.4 Evaluación de expresiones	Evaluación paso a paso de expresiones y conversiones.
2. Secuencias y Entrada/Salida	2.5 Ejercicios prácticos	Retos guiados para afianzar E/S y operadores.
3. Condicionales	3. Condicionales	Decisiones con if/else y estructuras alternativas.
3. Condicionales	3.1 Estructura if y if/else	Bloques condicionales y anidados.

3. Condicionales	3.4 Casos prácticos en cada lenguaje	Ejemplos equivalentes en C++, Java, JS y Python.
4. Ciclos	4. Ciclos	Iteración estructurada y control de bucles.
4. Ciclos	4.1 Ciclo for	Uso típico y variantes por lenguaje.
4. Ciclos	4.2 Ciclo while	Iteración basada en condición.
4. Ciclos	4.3 Ciclo do-while	Evaluación al final y casos de uso.
4. Ciclos	4.4 Control de bucles (break, continue)	Interrupción y salto dentro del ciclo.
4. Ciclos	4.5 Patrones comunes de iteración	Acumuladores, contadores y búsquedas.
5. Cadenas	5. Cadenas	Manipulación de texto y diferencias entre lenguajes.
5. Cadenas	5.1 Concatenación y manipulación básica	Operaciones comunes: longitud, concatenar y formatear.
5. Cadenas	5.2 Búsqueda y extracción de subcadenas	indexOf/find, substrings y slicing.
5. Cadenas	5.3 Parsing y análisis de texto	Split, tokenización y conversión de tipos.
5. Cadenas	5.4 Diferencias entre lenguajes	String vs char[], inmutabilidad y métodos.
6. Vectores / Arrays	6. Vectores / Arrays	Estructuras indexadas y uso eficiente.
6. Vectores / Arrays	6.1 Declaración y creación	Arreglos estáticos y dinámicos según lenguaje.
6. Vectores / Arrays	6.2 Acceso y recorrido	Indices, límites y patrones de recorrido.
6. Vectores / Arrays	6.3 Inserción y eliminación de elementos	Shift/unshift/splice; push/pop; conceptos en C++/Java.
6. Vectores / Arrays	6.4 Ordenamientos básicos (nociones)	Burbuja/selección/inserción (visión general).

		operaciones típicas.
7. Matrices	7.1 Arreglos bidimensionales	Declaración y memoria; listas de listas.
7. Matrices	7.2 Recorrido por filas y columnas	Patrones de acceso eficientes.
7. Matrices	7.3 Suma, promedio, máximos/mínimos	Agregación y estadísticas básicas.
7. Matrices	7.4 Transposición de matrices	Intercambio de filas/columnas y costo.
7. Matrices	7.5 Casos reales (tableros, mapas, tablas)	Aplicaciones prácticas de matrices.
8. Funciones y Procedimientos	8. Funciones y Procedimientos	Modularidad y reutilización de código.
8. Funciones y Procedimientos	8.1 Definición y llamada	Firma, nombre y llamada de funciones.
8. Funciones y Procedimientos	8.2 Parámetros y retorno	Por valor/referencia; tipos de retorno.
8. Funciones y Procedimientos	8.3 Alcance de variables (scope)	Local/global; sombras y tiempos de vida.
8. Funciones y Procedimientos	8.4 Sobrecarga y variaciones por lenguaje	Overloading en C++/Java, alternativas en JS/Python.
8. Funciones y Procedimientos	8.5 Buen diseño de funciones	Nombres claros, pureza y pruebas.
9. Listas	9. Listas	Estructuras enlazadas y casos de uso.
9. Listas	9.1 Concepto de nodo	Estructura Nodo: dato y punteros.
9. Listas	9.2 Inserción en lista simple	Insertar al inicio/final y en medio.
9. Listas	9.3 Eliminación de nodos	Eliminar por valor/posición y manejo de punteros.
9. Listas	9.4 Recorrido y búsqueda	Iteración y búsqueda lineal.
9. Listas	9.5 Implementación manual y usos comunes	Construcción desde cero y aplicaciones.

		memoria.
10. Pilas (Stack)	10.2 Operaciones: push, pop, peek	Definición y complejidad de operaciones.
10. Pilas (Stack)	10.3 Aplicaciones reales	Deshacer (CTRL+Z), historial de navegación, evaluación de expresiones.
11. Colas (Queue)	11. Colas (Queue)	Modelo FIFO y variaciones.
11. Colas (Queue)	11.1 Estructura FIFO	Concepto y representación en memoria.
11. Colas (Queue)	11.2 Operaciones: enqueue, dequeue	Definición y complejidad de operaciones.
11. Colas (Queue)	11.3 Colas circulares (noción)	Idea de cola circular y ventajas.
11. Colas (Queue)	11.4 Introducción a colas con prioridad	Concepto de prioridad y casos de uso.
12. Recorridos	12. Recorridos	Recorrido en estructuras lineales y jerárquicas.
12. Recorridos	12.1 Recorridos lineales (arrays, listas)	Iteración y patrones en estructuras lineales.
12. Recorridos	12.2 Recorridos jerárquicos (árboles)	Recorridos pre/in/postorden y niveles.
12. Recorridos	12.3 Introducción a BFS	Búsqueda en anchura: cola y niveles.
12. Recorridos	12.4 Introducción a DFS	Búsqueda en profundidad: pila/recursión.
13. POO	13. Programación Orientada a Objetos (POO)	Visión general y objetivos de POO.
13. POO	13.1 Conceptos básicos: clases y objetos	Definición de clase, objeto y estado/comportamiento.
13. POO	13.2 Encapsulación (atributos y métodos)	Ocultamiento de datos y acceso controlado.
13. POO	13.3 Constructores y destructores	Inicialización y liberación de recursos por lenguaje.

	polimorfismo	(noción).
13. POO	13.6 Aplicación práctica simple	Mini proyecto orientado a objetos en el lenguaje elegido.