



# Estructuras de Datos I

## Ejercicios

### Introducción a la Programación Orientada a Objetos

Cuando se solicite, escriba los archivos de C++ para ejecutar los programas que resuelvan cada uno de los requerimientos solicitados. Escriba la solución a los puntos en un archivo `.cpp` diferente y escriba en un archivo de texto la explicación de la forma en que resolvió cada uno de los puntos del ejercicio (use el número de la línea de código para referenciar instrucciones en el documento). Comprima los archivos `.cpp` y el documento de texto en un archivo `.zip` con el nombre **EjercicioClase3-<código estudiante>.zip**. En el campus virtual estará un enlace habilitado para que suba el archivo hasta el **domingo 4 de marzo a las 23:55**.

1. (No requiere archivo `.cpp`) Se tiene el siguiente fragmento de código escrito en C++:

```
int    foo = 10;  
int*   ptr = &foo;
```

¿Cuáles de las siguientes instrucciones modifica el valor de `foo` y lo vuelve 11?. Explique su respuesta.

- a) `ptr++`
- b) `foo++`
- c) `(*foo)++`
- d) `(*ptr)++`
- e) `(&foo)++`
- f) `(&ptr)++`

2. (No requiere archivo `.cpp`) Se tiene el siguiente fragmento de código en C++:

```
char c = 'T', d = 'S';
char *p1 = &c;
char *p2 = &d;
char *p3;
```

Asuma que la dirección donde esta almacenada `c` es 7721, y donde está almacenado `d` es 6925. ¿Qué mostrará la consola al ejecutar el siguiente bloque de código?. Explique su respuesta.

```
p3 = &d;
cout << " *p3 = " << *p3 << "\n"; // 1

p3 = p1;
cout << " *p3 = " << *p3 << "\n"; // 2
cout << " p3 = " << p3 << "\n"; // 3

*p1 = *p2;
cout << " *p1 = " << *p1 << "\n"; // 4
cout << " p1 = " << p1 << "\n"; // 5
```

3. (requiere archivo `.cpp`) Se requiere un programa para calcular las notas de los estudiantes de un curso basado en las siguientes consideraciones:

- Se tiene un arreglo con los nombres de los estudiantes, un arreglo con los códigos de los estudiantes, un arreglo con las notas del quiz1, un arreglo con las notas del quiz2, un arreglo con las notas del parcial y un arreglo con las notas del proyecto.
- Cada índice del arreglo corresponde a un estudiante y todos los arreglos deben tener el mismo tamaño, sino debe mostrar un mensaje y salir del programa. Use esta instrucción para salir (`exit(EXIT_FAILURE);`).
- Defina un arreglo que almacene las notas finales teniendo que la evaluación del curso es la siguiente:
  - quizzes: 25 %
  - parcial: 35 %
  - proyecto: 40 %
- Finalmente muestre en consola la lista de todos los estudiantes con todas las notas incluyendo la nota final

4. (requiere archivo `.cpp`) Se requiere realizar un programa que calcule la cantidad de veces que se repite un valor en un arreglo. El programa tiene un arreglo de algún tipo de dato (`char`, `float`, `int`) y una variable `buscar` del mismo tipo del arreglo. Tenga en cuenta que un arreglo de caracteres puede ser declarado e inicializado de la siguiente

forma:

```
char palabras[10] = "abcdefg hi";
```

Si los tipos de datos del arreglo y la variable no coinciden debe mostrar un mensaje y salir del sistema; use esta instrucción para salir (`exit(EXIT_FAILURE);`) y está para obtener el tipo de dato (`typeid(nombre.variable).name()`); para usar esta instrucción debe incluir la librería `typeinfo` (`#include<typeinfo>`). Se debe definir una variable donde se indique cuantas veces se encuentra el valor de la variable `buscar` en el arreglo e imprimirlo en pantalla.

5. (requiere archivo `.cpp`) Realice un programa que calcule la matriz traspuesta  $A^t$  de una matriz  $A$ . En una matriz traspuesta el elemento  $a_{ji}$  de la matriz original  $A$  se convertirá en el elemento  $a_{ij}$  de la matriz  $A^t$ . Una vez calculada muéstrela en pantalla.
6. (requiere archivo `.cpp`) Se requiere realizar un programa que realice las operaciones de suma y resta sobre una matriz  $A$  y una matriz  $B$ . Ambas matrices deben tener la misma dimensión (mismas filas y columnas en A y B) y se realiza la suma o resta componente a componente como se muestra en el ejemplo:

$$A = \begin{bmatrix} 5 & 3 & -9 & 7 \\ -2 & 7 & 6 & 10 \\ 3 & 9 & 5 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 8 & 4 & 6 & 1 \\ 1 & 9 & -2 & 6 \\ 3 & 10 & 8 & 3 \end{bmatrix}$$

$$A + B = \begin{bmatrix} 5+8 & 3+4 & -9+6 & 7+1 \\ -2+1 & 7+9 & 6+(-2) & 10+6 \\ 3+3 & 9+10 & 5+8 & 2+3 \end{bmatrix} = \begin{bmatrix} 13 & 7 & -3 & 8 \\ -1 & 16 & 4 & 16 \\ 6 & 19 & 13 & 5 \end{bmatrix}$$

$$A - B = \begin{bmatrix} 5-8 & 3-4 & -9-6 & 7-1 \\ -2-1 & 7-9 & 6-(-2) & 10-6 \\ 3-3 & 9-10 & 5-8 & 2-3 \end{bmatrix} = \begin{bmatrix} -3 & -1 & -15 & 6 \\ -3 & -2 & 8 & 4 \\ 0 & -1 & -3 & -1 \end{bmatrix}$$

El programa debe validar que las matrices tengan las mismas dimensiones sino debe mostrar un mensaje de error y salir del programa. Use esta instrucción para salir (`exit(EXIT_FAILURE);`). Para calcular el tamaño de las filas y las columnas use las siguientes instrucciones:

```
// tamaño total de arreglo de dos dimensiones(filas*columnas)
int size = sizeof(arreglo2D)/sizeof(arreglo2D[0][0]);

// número de columnas en el arreglo de dos dimensiones
int columnas = sizeof(arreglo2D[0])/sizeof(arreglo2D[0][0]);
```