TPN-4

**1.** Diferencia entre funciones y procedimientos en C++

Funciones: Devuelven un valor usando 'return'. Se usan cuando se necesita un resultado específico.  
Procedimientos: Son funciones tipo 'void', no devuelven valor. Se usan para ejecutar tareas.  
Ejemplo de función:  
int sumar(int a, int b) { return a + b; }  
Ejemplo de procedimiento:  
void mostrarMensaje() { cout << "Hola Mundo"; }

**2. Componentes de una función**

- Tipo de retorno  
- Nombre  
- Parámetros  
- Cuerpo  
Ejemplo: int sumar(int a, int b) { return a + b; }

**3. Diferencias entre arreglo de caracteres y string**

|  |  |  |
| --- | --- | --- |
| Característica | Arreglo de caracteres | string (STL) |
| Manejo | Manual | Automatizado |
| Terminación | Requiere \0 | No es necesario |
| Métodos | No | Sí |
| Ejemplo | char nombre[] = "Juan"; | string nombre = "Juan"; |

**4. Métodos de la clase string**

Tabla de métodos con ejemplo de uso:

|  |  |  |
| --- | --- | --- |
| Método | Descripción | Ejemplo |
| max\_size() | Máximo número de caracteres | s.max\_size(); |
| compare() | Compara con otra string | s1.compare(s2); |
| copy() | Copia en un char[] | s.copy(buffer, 4); |
| c\_str() | Devuelve const char\* | s.c\_str(); |
| data() | Puntero a datos internos | s.data(); |
| empty() | ¿Está vacía? | s.empty(); |
| erase() | Borra parte del string | s.erase(3,2); |
| find() | Busca substring | s.find("lo"); |
| find\_last\_of() | Última ocurrencia | s.find\_last\_of("aeiou"); |
| front() | Primer carácter | s.front(); |
| insert() | Inserta en posición | s.insert(5, "Hola "); |
| replace() | Reemplaza parte | s.replace(0,4,"Adiós"); |
| reserve() | Reserva memoria | s.reserve(100); |
| resize() | Cambia tamaño | s.resize(5); |
| substr() | Subcadena | s.substr(2,3); |
| swap() | Intercambia con otro | s1.swap(s2); |

**5. ¿Qué es un vector en C++?**

Es una estructura de la STL que almacena elementos dinámicamente.  
Características: tamaño dinámico, acceso aleatorio, métodos útiles.  
Ejemplo: vector<int> numeros = {1, 2, 3};

**6. ¿Cómo se modifica el tamaño?**

Usando resize(). Ejemplo: v.resize(5);

**7. ¿Qué pasa si accedés fuera de rango?**

Con [] es comportamiento indefinido, con at() lanza excepción.

**8. Clase vector**

Permite almacenar y manipular elementos con acceso aleatorio y tamaño dinámico.

**9. Métodos de vector**

|  |  |  |
| --- | --- | --- |
| Método | Descripción | Ejemplo |
| size() | Tamaño actual | v.size(); |
| capacity() | Capacidad interna | v.capacity(); |
| empty() | ¿Vacío? | v.empty(); |
| push\_back() | Agregar al final | v.push\_back(10); |
| pop\_back() | Eliminar último | v.pop\_back(); |
| front() | Primer elemento | v.front(); |
| back() | Último elemento | v.back(); |
| at() | Acceso seguro | v.at(2); |
| assign() | Reemplaza contenido | v.assign(otro.begin(), otro.end()); |
| erase() | Elimina uno o más | v.erase(v.begin()+1); |
| insert() | Inserta elementos | v.insert(v.begin(), 5); |
| clear() | Elimina todos | v.clear(); |
| swap() | Intercambia | v.swap(otro); |

**10. Clases y Objetos**

Una clase es una plantilla de código que define atributos y métodos.  
Ejemplo:  
class Persona {  
 public:  
 string nombre;  
 void saludar();  
};  
Un objeto es una instancia de esa clase. Ejemplo:  
Persona p;  
p.nombre = "Ana";