
Laboratorio 18

Funtores

1. Competencias

1.1. Competencias del curso

Conoce, comprende e implementa programas usando funtores del lenguaje de programación C++.

1.2. Competencia del laboratorio

Conoce, comprende e implementa programas usando funtores del lenguaje de programación C++.

2. Equipos y Materiales

- Un computador.
- IDE para C++.
- Compilador para C++.

3. Marco Teórico

3.1. Funtores

Los funtores son los objetos de clase que al instanciarlos pueden comportarse como funciones.

La principal ventaja de los funtores es que, aunque no son funciones, aún pueden guardar el estado ya que son objetos de clase.

Los funtores se hacen sobrecargando el operador () en cualquier clase para que el objeto de esa clase pueda ser llamado, como llamamos a una función simple.

Veamos el siguiente ejemplo para tener una comprensión clara.

```
class Bienvenida {  
public:  
    void operator()() {  
        cout << "Buen día alumnos" << endl;  
    }  
};
```

Ahora para usar esto, crearemos un objeto de la clase Bienvenida y lo llamaremos así:

```
int main()  
{  
    Bienvenida obj;  
    obj();  
    system("pause");  
    return 0;  
}
```

El resultado sería el siguiente:

Buen día alumnos

Podemos ver en la salida que el operador de llamada de función llamada operador() se llama cuando usamos el objeto con el operador.

3.1.1. Funtores con parámetros en C++

Los funtores también se pueden definir con algunos parámetros y tipo de retorno, al igual que las funciones. Podemos modificar el functor anterior para incluir el parámetro en él.

```
#include "stdafx.h"  
#include <iostream>  
using namespace std;  
  
class Bienvenida {  
public:  
    void operator()(string nombre) {  
        cout << "Buen día " << nombre << endl;  
    }  
};
```

Para llamar a esto, también debemos pasar el parámetro.

```
int main()
{
    Bienvenida obj;
    obj("ALumnos");
    system("pause");
    return 0;
}
```

El resultado sería el siguiente:

Buen día Alumnos

3.1.2. Funtores con parámetros y retorno en C++

Los funtores también se pueden definir con algunos parámetros y tipo de retorno, al igual que las funciones. Podemos modificar el functor anterior para incluir el parámetro y retorno.

```
#include "stdafx.h"
#include <iostream>
using namespace std;

class Bienvenida {
public:
    string operator()(string nombre) {
        string saludo = "Buen día " + nombre;
        return saludo;
    }
};
```

Para llamar a esto, también debemos pasar el parámetro.

```
int main()
{
    Bienvenida obj;
    string saludo = obj("ALumnos");
    cout << saludo << endl;
    system("pause");
    return 0;
}
```

El resultado sería el siguiente:

Buen día Alumnos

4. Ejercicios

Resolver los siguientes ejercicios planteados:

1. Utilizando Funtores, elabore una clase que calcule la ecuación de regresión lineal simple ($y = a + bc$) de un conjunto de pares de datos (x, y) almacenados en un vector, retorne como parte del resultado los valores de a y b . Apóyese en funtores para calcular las diferentes sumatorias que pueden presentarse.
2. Utilizando Funtores, elabore una clase que simule el proceso de la función estándar FIND. Se debe trabajar enviando como parámetros el índice de inicio, el índice final de la búsqueda y el dato a buscar. Retorne todas las ocurrencias iguales dentro del rango indicado (debe devolver un vector con los índices de todas las ocurrencias)
3. Utilizando Funtores y el método `std::sort`, elabore una clase Elementos con dos atributos de números enteros a y b . Genere una lista de 20 elementos de esta clase Elementos con valores aleatorios tanto para a y b . Mediante el método `std::sort` ordénelos de la forma en que un Objeto al ser comparado con un segundo se tenga la desigualdad : $\text{obj1.a} < \text{obj2.b}$. El método `std::sort` debe de trabajar con un objeto Funtores. De ser necesario, investigue como realizar este procedimiento que dependa de un objeto del tipo Funtores.

5. Entregables

Al final estudiante deberá:

1. Compactar el código elaborado y subirlo al aula virtual de trabajo. Agregue sus datos personales como comentario en cada archivo de código elaborado.
2. Elaborar un documento que incluya tanto el código como capturas de pantalla de la ejecución del programa. Este documento debe de estar en formato PDF.
3. El nombre del archivo (comprimido como el documento PDF), será su LAB19_GRUPO_A/B/C_CUI_1erNOMBRE_1erAPELLIDO.

(Ejemplo: LAB19_GRUPO_A_2022123_PEDRO_VASQUEZ).

4. Debe remitir el documento ejecutable con el siguiente formato:

LAB19_GRUPO_A/B/C_CUI_ EJECUTABLE_1erNOMBRE_1erAPELLIDO

(Ejemplo: LAB19_GRUPO_A_EJECUTABLE_2022123_PEDRO_VASQUEZ).

En caso de encontrarse trabajos similares, los alumnos involucrados no tendrán evaluación y serán sujetos a sanción.