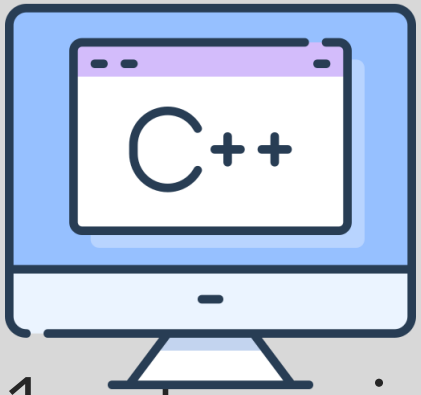
TALLER DE EJERCICIOS EN C++ AVANZADO.

PROTECO 2023-1

Talleres.



Temas para C++ avanzado.

1. Funciones.
2. Manejo de memoria dinámica.
3. Manejo de bibliotecas básicas.
4. Sobrecarga de operadores.
5. Estructura de datos en C++.
    1. Manejo de apuntadores.

Extra.

1. Matemáticas nivel secundaria bachillerato

Temática.

- Como es nivel avanzado de C++ se tomará nada más 2 ejercicios saltando la teoría básica.
- El ejercicio 1. Corresponderá a unos 30 minutos a 1 hora.
- El ejercicio 2. Corresponderá al resto de la sesión.
- Se dará al final la solución. (No son tan triviales los ejercicios).
- Se puede hacer en equipo o individual.

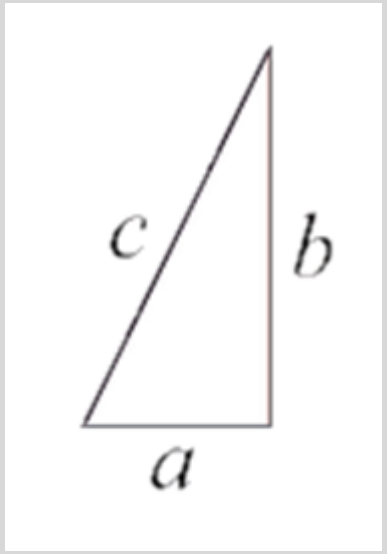
Ejercicio 1. “It’s All Greek To Me”.

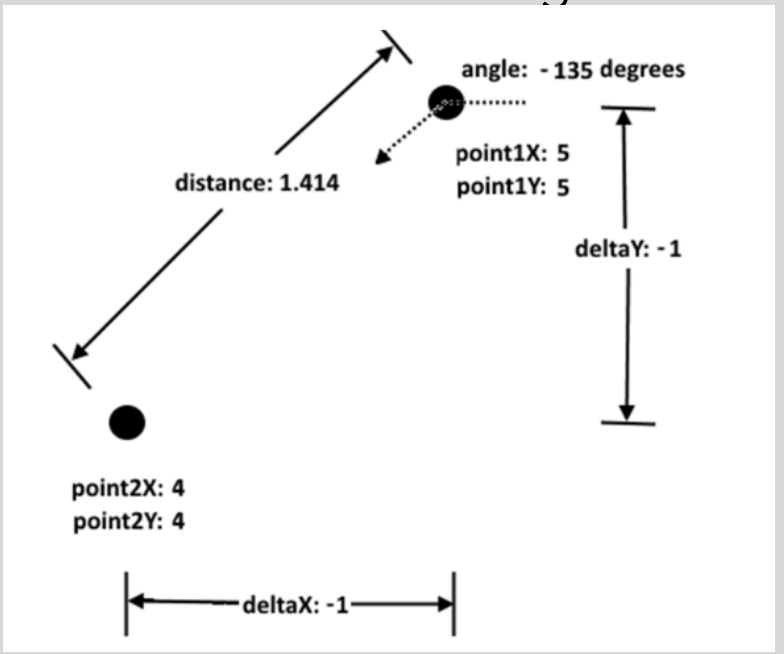
- It’s all greek to me significa en español “Esto está en chino”.

Imagina que has construido una maquina de tiempo en el que puedes viajar a través del tiempo a la época de los Griegos (Especialmente, Pitágoras y Hiparco) quien ha contribuido asombrosos teoremas (Teorema de Pitágoras) y las ramas de las matemáticas (Trigonometría). El ejercicio consiste calcular la distancia entre dos puntos y el ángulo un personaje que necesitará moverse de un punto a otro punto.

El teorema de Pitágoras.

Nos dice cómo calcular lo largo de la hipotenusa de un cierto ángulo.





Sabemos que el teorema dice que:

$$c^2 = a^2 + b^2$$

Entonces si tomamos la raíz cuadrada de $a^2 + b^2$, obtenemos c, el largo de la hipotenusa.

- Definir dentro de Main() una variable tipo **string** llamada **Input**.
- Usar **getline** que reciba la entrada por el usuario y también la variable **Input.**
- Opcional. Pueden definir los puntos como variables globales.
- Realizar una función tipo **void** donde el flujo de entrada de datos sea **Input.**
- En la función **void** tendrá que devolver los puntos en forma dado el string separarlos y además devolverlos para ser usado en las operaciones necesarias. (Sugerencia: Use un sobrecarga **stof** y también manejo de la biblioteca **string**.
- Dentro de **main** se manejará un ciclo **While** en donde mientras que la entrada sea distinta de **q** estará leyendo 4 puntos y haciendo las operaciones anteriores para obtener la **distancia** y **ángulo**, este último tendrá que estar en grados.
- Si se teclea **q** termina la ejecución del programa.

Sugerencias:

1. Usar **math** y para el obtener el ángulo **atan2**(double a, double b) en donde el resultado lo devuelve en radianes.
2. Usar M_PI para obtener el valor de PI más aproximado.
3. Para convertir de radianes a grados:

$$\theta = \alpha * \left(\frac{180}{\pi}\right)$$

Donde Alpha es el grado en radianes.

Test.

```
5 5 4 4
1.41421 -135
2 2 4 4
2.82843 45
0 0 1 0
1 0
0 0 0 1
1 90
0 0 -1 0
1 180
2 2 -4 4
6.32456 161.565
2 2 4 -4
6.32456 -71.565
6 5 4 3
2.82843 -135
3 4 5 6
2.82843 45
q


...Program finished with exit code 0
Press ENTER to exit console.
```

5 5 4 4

2 2 4 4

0 0 1 0

0 0 0 1

0 0 -1 0

2 2 -4 4

2 2 4 -4

6 5 4 3

3 4 5 6

Ejercicio 2. Estructura Bancaria de un nodo.

Se necesita por alguna extraña razón un nodo en lugar de una lista, cola, pila almacenar los siguientes datos bancarios.

- Calle
- País
- Estado
- Numero
- Nombre
- Apellido
- Edad.
- Cuenta.

1. Tener una estructura llamada **datos** o **gdata** en donde se tendrá los tipo que están en azul. No es necesario usar Typedelf para esta estructura.
2. Tener otra estructura (principal) llamada **client** donde estará los datos en verde, la diferencia es que también esta estructura almacenará la estructura de **gdata** o **datos**, es decir, que son estructuras anidadas. (Opcional, usar typedelf.)
3. El programa no tiene que tener variables **globales**.
4. Todas las funciones que se vayan a usar que no afecten a la estructura debe ser **void**, y no tienen que usar **return**. (Cuando se requiera manipular los datos la sugerencia es usar **apuntadores** o **pasos por referencia**.
5. Usar archivos de cabecera: **main.cpp, functions.h, functions.cpp**. Cuidado porque en C++´cambia un poco el uso de headers.
6. En **functions.h** debe estar las estructuras y los prototipos, además en **functions.cpp** lo que tienen que hacer las siguientes funciones:
    1. Creación de los clientes (Se llama a la función para que cree el espacio dinámicamente donde reciba como parámetro el valor de la siguiente función. Retorna una estructura.
    2. Función llamada **reserve** donde se crea dinámicamente y reciba el tamaño o el número de clientes. Retorna una estructura.
    3. Crear la función donde pide el número de clientes pero esta debe que se capaz de guardar el valor de los clientes y usarlos en las funciones 1 y 2 sin **return**.
    4. Crear una función tipo estructura, en donde se llenan todos los datos con el número de clientes. (Retorna el nodo o la estructura con todos los clientes.)
    5. En una función debe imprimir el estado del cliente que en este caso solamente debe ser el **no. Cliente, nombre, apellido, su dinero en pesos mexicanos, dólares, euros y yenes (Japón)**.
    6. Crear una función donde aparezca los datos del cliente los personales.
    7. Declarar mediante un **template** o plantilla una función donde pueda escribir en un archivo(s) el estado del cliente o de los clientes o sus datos personales, el administrador tiene que dar el nombre del archivo y el tipo de operación que se va realizar.
    8. Crear una función donde imprima todos los datos del cliente o clientes.
    9. Crear una función en donde borre todos los datos del cliente y se libere la memoria usada.
7. En el **main.cpp** solamente debe aparecer una función llamada **menú** () dentro de la función **main** en la cual no tiene que recibir ningún parámetro.

Sugerencias.

1. Usar **pragma, #include <string.h>, #include<fstream>.**

Test.

```
Number of clients: 2
Uploading...
Process successfully
                PROTECO Bank... (ADMINISTRATOR)
Options
Create new clients. (Default)
1. Upload data.
2. Status Client.
3. Print data.
4. Print on a file data.
5. Delete clients (All) and exit
===> 1
Client #1
Name: John
Surname: Becker
Edge: 19
Saved: 1687
Contry: Mexico
State(Without spaces, with -): CDMX
Street(Without spaces, with -): Rivera
Number: 79

Client #2
Name: Yoshio
Surname: Keima
Edge: 22
Saved: 798465
Contry: Japan
State(Without spaces, with -): Nara
Street(Without spaces, with -): Ahoyama
Number: 165
```

```
                PROTECO Bank... (ADMINISTRATOR)
Options
Create new clients. (Default)
1. Upload data.
2. Status Client.
3. Print data.
4. Print on a file data.
5. Delete clients (All) and exit
===> 2
                                                Status client...Change and Change.
Client #1
Name: John
Surname: Becker
MXN: 1687
USD: 84.35
EUR: 86.037
Yen: 12500.7
                                                Status client...Change and Change.
Client #2
Name: Yoshio
Surname: Keima
MXN: 798465
USD: 39923.3
EUR: 40721.7
Yen: 5.91663e+06
```

```
                PROTECO Bank... (ADMINISTRATOR)
Options
Create new clients. (Default)
1. Upload data.
2. Status Client.
3. Print data.
4. Print on a file data.
5. Delete clients (All) and exit
===> 3
Client: 1
Name: John
Surname: Becker
Edge: 19
Saved: 1687
Contry: Mexico
State: CDMX
Street: Rivera
Number: 79

Client: 2
Name: Yoshio
Surname: Keima
Edge: 22
Saved: 798465
Contry: Japan
State: Nara
Street: Ahoyama
Number: 165
```

```
                PROTECO Bank... (ADMINISTRATOR)
Options
Create new clients. (Default)
1. Upload data.
2. Status Client.
3. Print data.
4. Print on a file data.
5. Delete clients (All) and exit
===> 4
What do you want to file generate MD[0] or ST[1]? 0
File's name: users.txt
File(s) exported...             PROTECO Bank... (ADMINISTRATOR)
Options
Create new clients. (Default)
1. Upload data.
2. Status Client.
3. Print data.
4. Print on a file data.
5. Delete clients (All) and exit
===> 4
What do you want to file generate MD[0] or ST[1]? 1
File's name: data_users.txt
File(s) exported...             PROTECO Bank... (ADMINISTRATOR)
```

```
File(s) exported...                PROTECO Bank... (ADMINISTRATOR)
Options
Create new clients. (Default)
1. Upload data.
2. Status Client.
3. Print data.
4. Print on a file data.
5. Delete clients (All) and exit
===> 5


...Program finished with exit code 0
Press ENTER to exit console.
```

main.cpp | functions.h | functions.cpp | users.txt | data_users.txt

```
                    Status client...Change and Change.
Client #1
Name: John
Surname: Becker
MXN: 1687
USD: 84.35
EUR: 86.037
Yen: 12500.7

                    Status client...Change and Change.
Client #2
Name: Yoshio
Surname: Keima
MXN: 798465
USD: 39923.3
EUR: 40721.7
Yen: 5.91663e+06

```

main.cpp | functions.h | functions.cpp | users.txt | data_users.txt

```
                    Status client...Data and Data.
Client: 1
Name: John
Surname: Becker
Edge: 19
Saved: 1687
Contry: Mexico
State: CDMX
Street: Rivera
Number: 79

                    Status client...Data and Data.
Client: 2
Name: Yoshio
Surname: Keima
Edge: 22
Saved: 798465
Contry: Japan
State: Nara
Street: Ahoyama
Number: 165
```