CECS 2200-20 p1c8

S6: Lunes 14 abril	• *Variables (basic types, declaration, initialization, scope rules, casting), constants, literal, keywords* • *Standard library* • *Arithmetic operators (+, -, *, /, %), precedence, and association* • *Arithmetic expressions* • *Math Library*

DECLARACION E INICIALIZACION DE VARIABLES<<<<<<<<<<<<
Basic types:
int myInt; // estamos declarando una variable de tipo entero con nombre myInt.

float myFloat; // estamos declarando una variable tipo floating point (decimal) con nombre myFloat

double myD; // declarando una variable tipo double (doble precisión) con nombre myD

char myC; // declarando una variable tipo caracter con nombre myC

bool myB; // declaro una variable tipo booleano (true false) con nombre myB

//Para inicializar se hace
myInt = 0;
myFloat = 3.14;

//Se puede declarar e inicializar a la vez
int myInt=0; //declaro e inicializo a la vez.

//Si tienen varias variables, se pueden declarar e inicializar todas (son del mismo tipo) en una linea separadas por comas!!

int my1=0, my2, my3=90, my4;

SCOPE RULES

Scope es el trozo de codigo donde una variable es válida!!!

Una variable es válida, despues de declararla hasta el próximo }

Despues del } la variable ya no está disponible!!!

Excepcion: variables globales, que son declaradas (se dice definidas) antes del main!!!

```
1   #include <iostream>
2   using namespace std;
3
4   int main(){
5       int a=10,b=20;
6       {
7           int a=1,b=2;
8       }
9       cout << "a: " << a << " b: " << b << endl;
10      return 0;
11  }
```

```
1   #include <iostream>
2   using namespace std;
3
4   int main(){
5       int a=10,b=20;
6       {
7           cout << "dentro del scope a: " << a << " b: " << b << endl;
8           int a=1,b=2;
9           cout << "dentro del scope a: " << a << " b: " << b << endl;
10      }
11      cout << "fuera del scope a: " << a << " b: " << b << endl;
12      return 0;
13  }
```

```
C:\MinGW\bin>g++ main.cpp -o main.exe

C:\MinGW\bin>main
dentro del scope a: 10 b: 20
dentro del scope a: 1 b: 2
fuera del scope a: 10 b: 20

C:\MinGW\bin>_
```

CASTING

Cambiar un type def por otro!!!

//Por ejemplo, tengo dos enteros
int a=1,b=2;
float c;
//quiero dividir a entre b
c = a/b; //El resultado es 0.0 porque la division es entera, antes de la asignacion.
c = (float)a/b; //El resultado es 0.5 porque hice un CASTING!!!

```
1   #include <iostream>
2   using namespace std;
3
4   int main(){
5       int a=1,b=2,d;
6       float c;
7       cout << "resultado de la division entera: " << a/b << endl;
8       c = a/b;
9       cout << "resultado de la division entera y asignada a float: " << c << endl;
10      cout << "resultado de la division CASTEADA entera: " << (float)a/b << endl;
11      c = (float)a/b;
12      cout << "resultado de la division CASTEADA entera y asignada a float: " << c << endl;
13      d = (float)a/b;
14      cout << "resultado de la division CASTEADA entera y asignada a entero: " << d << endl;
15
16
17      return 0;
18  }
```

```
C:\MinGW\bin>g++ main.cpp -o main.exe

C:\MinGW\bin>main
resultado de la division entera: 0
resultado de la division entera y asignada a float: 0
resultado de la division CASTEADA entera: 0.5
resultado de la division CASTEADA entera y asignada a float: 0.5
resultado de la division CASTEADA entera y asignada a entero: 0

C:\MinGW\bin>_
```