**HARVARD CS50x**

argumento → função → efeito

Em C, temos varias bibliotecas, que englobam varias funções e comandos.

Standard Library → #include <stdio.h>

* Eu estou usando a <cs50.h> também neste curso

string answer = get\_string("What's your name? \\n");

printf ("Hello! %s \\n", answer);

= atribuição

== semelhança

contador = contador + 1; → contador += 1; → contador++;

int n1 = get\_int("what's the 1st number?");

int n2 = get\_int("And the second?");

if(n1 < n2)

{

printf("n1 < n2");

}

else if (n1 > n2)

{

printf("n1 > n2");

}

else if (n1 == n2)

{

printf("n1 == n2");

}

Usando o else if, seu código fica mais fluído, melhora o design…

Fluxograma da lógica aplicada.

Operador OR → || (se *p* então *q* ou *r*) → Apenas um dos dois deve ser verdadeiro

if (c == 'y' || 'Y')

{

printf("agreed!");

}

Operador AND → && (se *p* então *q* e *r*) → Os dois deve ser verdadeiro

CONTADOR (negativo)

int count = 3;

while (count > 0)

{

printf("Hello! \\n");

count--;

}

**Functions**

→ você precisa definir funções antes de usa-las

void meow (void) // Fazendo isso, você cria sua própria função

{

printf("Meow! \n"); }

function prototype

… printf(”meow!\n”);

}

…

→ Observe que ele declarou void meow (void), na linha 3 e então só executou na linha 13.

indica ao compilador que a função ainda será criada.

OBSERVAÇÃO IMPORTANTE.

Você tem que declarar a variável na sua função secundária. Mesmo que ela esteja na principal, não funciona

nesse caso, por exemplo nao iria funcionar, pq não declarou “x” ou “y”.

→ Isso se chama **escopo**, que é o contexto em que as variáveis existem.

Para resolver isso… (Se não entender, Lecture 1 - 1h37m45s)

const quer dizer que o valor **não pode mudar** depois — será sempre o número que foi definido

Exemplo de do{ } e while()…

***Type Casting*** - trata um tipo de dado como se fosse outro tipo

Se está dividindo dois inteiros, que tenham resto, ele corta tudo depois da virgula e só deixa o valor antes da virgula

1/3 = 0

para isso, colocasse “(float)” antes da variável…

**printf(".6%f \n", z);** - O uso do .6 diz quantas casas decimais terá.