**Comandos C++**

**sizeof =** Retorna o tamanho(em bits) do tipo da variavel (Aula 18);

**‘\0’ =** Código do enter (Aula 20);

**Int Argc =** Guarda o número de parâmetros informados (Aula 21);

**Char argv\*[] =** Armazena os argumentos; Ponteiro para uma Matriz de Ponteiros de Caracteres (21);

**strcmp() =** Faz comparação de duas Strings (Aula 21);

***Aula 27 - enum:***

**-** Conjuntos de constantes inteiras, que especificam valores que uma variavel desse tipo pode ter:

//se não declarar o valor dos itens, o valor começa em 0, proximo 1, proximo 2, ...

//Quando declarando um valor, e o próximo item não, esse item recebe o valor anterior +1

//isso se aplica a qualquer valor se não declarado no seguinte

***Aula 28 - Pilha/Stack Parte 1:***

**-** O Primeiro elemento a Entrar, é o último a Sair;

**-** O Último elemento a Entrar é o Primeiro a Sair;

- O último elemento a Entrar é o Primeiro da Stack;

**push() =** Insere o elemento;

**size() =** Verifica otamanho do elemento;

**top() =** Retorna o elemento que esta no TOPO (ultimo a ser inserido)**;**

**pop() =** Remove do TOPO elemento da Stack;

***Aula 29 - Pilha/Stack Parte 1:***

**empty() =** Verifica se a Pilha esta vazia, retorna TRUE ou FALSE;

***Aula 30 - Fila/Queue:***

**-** O Primeiro elemento a Entrar, é o Primeiro a Sair;

**-** O Último elemento a Entrar é o Último a Sair;

**front() =** Verifica quem está na frente da Queue (primeiro a ser inserido);

**back() =** Verifica quem está no final da Queue (ultimo a ser inserido;

***Aula 31 – List Parte 1:***

**push\_front() =** Insere pela frente/inicio da List;

**push\_back() =** Insere a Trás/Final da List;

**front() =** Retorna o elemento da Frente/Inicio;

**back() =** Retorna o elemento de Trás/Final;

***::interator = Forma de repetir um bloco de comandos;***

**begin() =** Inicio da List;

advance(ITERATOR, DISTANCIA) = Faz o avanço para a posição;

**insert(ITERETOR, VALOR) =** Insere o valor na posição;

**sort() =** Ordena;

**reverse() =** Inverte a ordem;

**pop\_front() =** Remove na Frente/Inicio;

***Aula 32 – List Parte 1:***

**eraser(POSIÇÃO) =** Remove o elemento da onde quiser;

**clear() =** Remove todos os elementos da List;

**list1.merge(list2) =** Mescla duas LISTAS, List1 recebe List2, e List2 fica vazia;

***Aula 33 - Struct:***

**- Coleção de variáveis, com propriedades e métodos (funções);**

***Aula 36 – Ponteiros***

**Ponteiros: armazena o endereço de uma Variavel do mesmo tipo;**

**\*= É o Ponteiro; & = É o endereço Endereço;**

***Aula 37 – Locação Dinâmica da Memória***

**- Armazena apenas o tamanho exato;**

**gets(VAR) =** Funciona como o CIN, porém ele armazena todo o conteúdo digitado;

**malloc( Tam Tipo VAR ) =** *(VOID)* Armazena apenas o tamanho exato;

***Aula 40 – Printf e Scanf; Requer <stdio.h>***

**Printf() =** Usado como o ***COUT***;

**Scanf() =** Usado como o ***CIN;***

***Aula 41 – Mais sobre Printf e Cout***

***Setbase(BASE) =*** *Converte o valor para base desejada;*

***Precision(NUM) =*** *Numero de casas decimais;*

***Setw(NUM) =*** *Tamanho que o elemente terá “espaços”;*

***Setfill(CONTEUDO) =*** *Conteúdo para preenchimento;*

***Aula 42 –Vector* <vector>**

***Vetor: São array unidimensionais;  
Vector: Estrutura mais moderna, com métodos bem elaborados;***

***Swap(VECTOR2) =***Faz a troca de valores de 2 vectores;

***At(POSIÇÃO OU OPERAÇÃO P/POSIÇÃO) =*** Retorna a posição escolhida do vector;

***Begin() =*** Insere no inicio;

***End() =*** Insere no final;

***Insert() =*** Insere na posição selecionada;

***Erase() =*** Remove na posição selecionada;

***Pop\_back() =*** Remove do final;

***Aula 46 – POO Parte 3 – Get e Set***

***Construtor: Método que é chamado quando se instância um objeto;***

***Private: Só podem ser acessados dentro da própria classe;***

***Get = Obtém o valor;***

***Set = Atribui um valor;***

**Bibliotecas**

**<stdlib.h> = Permite usar os comandos pause, e cls;**

**<stack> = Permite uso de Pilha/Stack;**

**<queue> = Permite o uso de Fila/Queue;**

**<list> = Permite o uso de Lista/List;**

**<stdio.h> = (34-40) Permite o uso da função gets() e os métodos Prinf e Scanf;**

**<stdlib.h> = (37) Permite o uso da função malloc;**

**<math.h> = (41) Biblioteca Matemática;**

**<iomanip> = (41) Manipulação ENT/SAI;**

**<vector> = (42) Permite o uni de Vectores;**