

## Aula prática 11

---

Esta aula tem como objetivo introduzir conceitos relacionados com a linguagem de programação C++ e programação orientada a objetos.

---

1. Considere um banho de 15 minutos em um chuveiro elétrico de potência de 5200W.
  - a) Construa a função `transforma_potencia(int w)` que deverá passar a potência do chuveiro de W para KW.
  - b) Construa a função `transforma_tempo(float min)`, que deverá transformar o tempo de minutos para horas.
  - c) Com a potência em kW e o tempo em horas, construa a função `calcula_consumo(float kw, float hora)` que deverá calcular o consumo de energia elétrica. Para obter esse consumo, usaremos a seguinte fórmula:  $kw * hora$ .

Depois de implementadas as funções, o programa deverá apresentar:

```
Horas = 0.25h
Potencia = 5.2kW
Consumo = 1.3KWh
```

2. Dado um gerador de números aleatórios de tamanho  $n$ , implemente as seguintes funcionalidades dentro do método `main()`:
  - a) Ordenar o [vetor](#) `v` e imprimir o vetor ordenado. Obs: Utilize o método `sort()`.
  - b) Remover o primeiro elemento e imprimir o novo tamanho do vetor.

Depois de implementadas as funcionalidades, o programa deverá apresentar:

```
Vetor Inicial
2 14 10 14 19 5 4 0 10 19
Vetor Ordenado
0 2 4 5 10 10 14 14 19 19
Novo tamanho: 9
Vetor final
2 4 5 10 10 14 14 19 19
```

3. Neste exercício pretende-se criar um programa de gestão de alunos em C++, utilizando o conceito de classes. O programa terá o nome GAC (Gestão de Alunos com Classes) e a introdução de dados para teste do programa poderá ser feita com recurso à edição da rotina `main()`. É possível encontrar dentro da pasta `aluno` um esqueleto de implementação, que deverá ser complementado.

- a) Pretende-se que qualquer objeto-instância da classe `Aluno` seja criado com um nome. Acrescente à classe `Aluno` um construtor que inicialize o atributo `nome` com o argumento que lhe for passado e que inicialize todos os outros atributos com valores nulos (0 para números, e "" para strings). O protótipo será:

```
Aluno(string nom);
```

Implemente também um outro construtor que inicialize todos os atributos com os valores fornecidos como argumentos (exceto `media`). Note que isto constitui um exemplo de sobrecarga de funções. O protótipo será:

```
Aluno(string nom, string cur, int num);
```

- b) Um aluno, durante o tempo de estudante, poderá mudar de curso e número algumas vezes, mas nunca de nome, e a sua média também poderá sofrer alterações. Escreva os membros-função que permitam efectuar as atribuições ou alterações permitidas (e.g. `setCurso(string cur)`) e escreva também as restantes funções de acesso típicas (e.g. `float getMedia()`).

Utilizando as funções de acesso que implementou, defina o curso (e.g. "Medicina") e o número (e.g. "2020123") do aluno `a1` antes da sua impressão. Defina também valores para as médias de todos os alunos, antes da sua impressão.

- c) Implemente o membro função

```
void imprimir(ostream &os) const;
```

que imprime na *stream* de saída os, que poderá ser um ficheiro ou o ecrã, as informações de um determinado aluno, obedecendo ao seguinte formato:

```
<nome> | <curso> | <numero> | <media>
```

Nota: exemplo de utilização (a definir na rotina `main()`).

```
Aluno al("Alberto Mateus", "Engenharia", 2019197);  
al.imprimirAluno(cout);
```

- d) Implemente o membro-função

```
string sigla() const;
```

que retorna a sigla de um determinado aluno, formada pelas iniciais do seu nome. Por exemplo, o aluno `Pedro Antunes Rocha` terá a sigla "PAR".