

 PUC Minas	Curso: Ciência da Computação	
	Disciplina: Algoritmos e Estrutura de Dados I	
	Nome: Pedro Henrique Lopes Costa	
	Código de pessoa: 1362465	Campus: Coração Eucarístico
	Período: 1º	Turma: Manhã

Questão 01

try: Método usado para testar uma função/classe, e, em caso de exceção, irá retornar um erro para ser tratado pelo catch a seguir.

catch: Método usado para tratar o erro e identificar visualmente pela string do erro ou passar. Em caso de loop, usar while.

finally: C++ não tem finally porque não é necessário. finally é usado para executar algum código independentemente de ter ocorrido uma exceção ou não, o que quase sempre é algum tipo de código de limpeza.

Questão 02

```
#include <iostream>
#include <math.h>
#include <locale.h>
#include <stdexcept>

using namespace std;

//
----- //

int main()
{
    setlocale(LC_ALL, "Portuguese");

    // ----- //

    while(1)
    {
        try
        {
            float sf;

            printf("> Digite um número inteiro: ");
            scanf("%f", &sf);

            if(sf != (int)sf) throw invalid_argument("O número inserido não é inteiro!");
            else return printf("O número inserido é inteiro.\n");
        }
        catch(invalid_argument x)
        {
            printf("%s\n\n", x.what());
        }
    }

    // ----- //

    system("pause");
    return 0;
}
```

Questão 03

```
#include <iostream>
#include <math.h>
#include <locale.h>
#include <stdexcept>

using namespace std;

class ExcecaoAcimaDeCem: public exception
{
public:

    string errorMsg() { return "A soma chegou a 100!"; }
};

// ----- //

int main()
{
    setlocale(LC_ALL, "Portuguese");

    // ----- //

    float soma = 0;
    int total = 0;

    while(1)
    {
        try
        {
            float sf;

            printf("> Digite um número para somar: ");
            scanf("%f", &sf);

            soma += sf;
            total++;

            printf("> Soma atual: %.1f\n> Total de números somados: %i\n> Média: %.1f\n\n", soma,
total, soma / total);

            if(soma > 100) throw ExcecaoAcimaDeCem();
        }
        catch(ExcecaoAcimaDeCem x)
        {
            printf("%s\n", x.errorMsg().c_str());
            return false;
        }
    }

    // ----- //

    system("pause");
    return 0;
}
```

Questão 04

```
#include <iostream>
#include <math.h>
#include <locale.h>
#include <stdexcept>

using namespace std;

class NumberFormatException: public exception
{
    public:

    string errorMsg() { return "Você deve inserir números inteiros!"; }
};

class ArrayIndexOutOfBoundsException: public exception
{
    public:

    string errorMsg() { return "O index inserido está fora da array!"; }
};

bool isNumber(const string& s)
{
    for(char const &ch : s)
    {
        if(std::isdigit(ch) == 0) return false;
    }
    return true;
}

//
-----
----- //

int main()
{
    setlocale(LC_ALL, "Portuguese");

    // ----- //

    int vetor[10];

    for(int i = 0; i != 10; i++)
    {
```

```

    string inserir, posicao;

    while(1)
    {
        try
        {
            printf("> Digite um número inteiro para inserir: ");
            cin >> inserir;

            if(!isNumber(inserir)) throw NumberFormatException();
            else break;
        }
        catch(NumberFormatException x)
        {
            printf("%s\n\n", x.errorMsg().c_str());
        }
    }

    while(1)
    {
        try
        {
            printf("> Digite a posição a inserir: ");
            cin >> posicao;

            printf("\n");

            int i_posicao;

            if(!isNumber(posicao)) throw
ArrayIndexOutOfBoundsException();
            else
            {
                i_posicao = stoi(posicao);

                if(i_posicao < 0 || i_posicao > 9) throw
ArrayIndexOutOfBoundsException();
                else
                {
                    vetor[i_posicao] = stoi(inserir);
                    break;
                }
            }
        }
        catch(ArrayIndexOutOfBoundsException x)

```

```
        {  
            printf("%s\n\n", x.errorMessage().c_str());  
        }  
    }  
}  
  
for(int x = 0; x != 10; x++) printf("vetor[%i] = %i\n", x, vetor[x]);  
  
printf("\n");  
  
// ----- //  
  
system("pause");  
return 0;  
}
```

Questão 05

```
#include <iostream>
#include <math.h>
#include <locale.h>
#include <stdexcept>

using namespace std;

//
-----
----- //

int main()
{
    setlocale(LC_ALL, "Portuguese");

    // ----- //

    double ponto[3];

    try
    {
        printf("> Digite o valor do ponto A: ");
        scanf("%lf", &ponto[0]);

        printf("> Digite o valor do ponto B: ");
        scanf("%lf", &ponto[1]);

        printf("> Digite o valor do ponto C: ");
        scanf("%lf", &ponto[2]);

        if(ponto[0] + ponto[1] <= ponto[2] || ponto[0] + ponto[2] <= ponto[1]
|| ponto[1] + ponto[2] <= ponto[0]) throw invalid_argument("Os pontos
informados não podem formar um triângulo!");
        else return printf("\n> Área do triângulo = %.2lf\n", (ponto[0] *
ponto[2]) / (double)2);
    }
    catch(invalid_argument x)
    {
        printf("\n%s\n\n", x.what());
    }

    // ----- //

    system("pause");
    return 0;
}
```