

1- Write a program to demonstrate the use of try, throw, and catch for handling division by zero

اكتب برنامجاً يوضح استخدام المحاولة والرمي والالتقاط للتعامل مع القسمة على صفر

Input

```
Enter numerator: 5
Enter denominator: 0
```

Output

```
Error: Division by zero is not allowed.
```

Solution

```
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    int numerator, denominator;

    cout << "Enter numerator: ";
    cin >> numerator;

    cout << "Enter denominator: ";
    cin >> denominator;

    try {
        if (denominator == 0) {
            throw "Division by zero is not allowed.";
        }

        cout << "Result: " << static_cast<double>(numerator) / denominator << endl;
    }
    catch (const char* errorMsg) {
        cout << "Error: " << errorMsg << endl;
    }

    return 0;
}
```

2- Write a program that takes two numbers as input, performs division, and handles potential exceptions for division by zero or negative input values. Print an error message if any exception occurs.

اكتب برنامجًا يأخذ رقمين كمدخلات، ويجري القسمة، ويتعامل مع الاستثناءات المحتملة للقسمة على صفر أو قيم مدخلة سالبة. اطبع رسالة خطأ في حالة حدوث أي استثناء.

Input

```
Enter first number: 1
Enter second number: 0
```

Output

```
Error: Division by zero is not allowed.
```

Solution

```
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    int num1, num2;

    cout << "Enter first number: ";
    cin >> num1;

    cout << "Enter second number: ";
    cin >> num2;

    try {
        if (num2 == 0) {
            throw "Division by zero is not allowed.";
        }

        if (num1 < 0 || num2 < 0) {
            throw "Input values must be non-negative.";
        }

        cout << "Result: " << num1 / num2 << endl;
    } catch (const char* errorMsg) {
        cout << "Error: " << errorMsg << endl;
    }

    return 0;
}
```

3- Write a program that takes a user's age as input and handles potential exceptions for invalid age values. Print an error message if any exception occurs.

اكتب برنامجًا يأخذ عمر المستخدم كمدخل ويتعامل مع الاستثناءات المحتملة لقيم العمر غير الصالحة. اطبع رسالة خطأ في حالة حدوث أي استثناء.

Input

```
Enter your age: 20
```

Output

```
Your age is: 20
```

Solution

```
// www.gammal.tech

#include <iostream>

using namespace std;
int main() {
    int userAge;

    printf("Enter your age: ");
    cin>>userAge;

    try {
        if (userAge < 0 || userAge > 120) {
            throw "Invalid age. Please enter a valid age between 0 and 120.";
        }

        printf("Your age is: %d\n", userAge);
    } catch (const char* errorMsg) {
        fprintf(stderr, "Error: %s\n", errorMsg);
    }

    return 0;
}
```

4- Write a program that takes a lowercase letter as input and handles potential exceptions for invalid input. Print an error message if any exception occurs.

اكتب برنامجاً يأخذ حرفاً صغيراً كمدخل ويتعامل مع الاستثناءات المحتملة للإدخال غير الصالح. اطبع رسالة خطأ في حالة حدوث أي استثناء.

Input

```
Enter a lowercase letter: A
```

Output

```
Error: Invalid input. Please enter a lowercase letter.
```

Solution

```
// www.gammal.tech

#include <iostream>
#include <cctype>

using namespace std;

int main() {
    char userInput;

    cout << "Enter a lowercase letter: ";
    cin >> userInput;

    try {
        if (!islower(userInput)) {
            throw "Invalid input. Please enter a lowercase letter.";
        }

        cout << "You entered: " << userInput << endl;
    } catch (const char* errorMsg) {
        cout << "Error: " << errorMsg << endl;
    }

    return 0;
}
```

5- Write a program that takes a user's grade as input (within the range 0-100) and handles potential exceptions for invalid grades. Print an error message if any exception occurs.

اكتب برنامجًا يأخذ تقدير المستخدم كمدخل (ضمن النطاق 0-100) ويتعامل مع الاستثناءات المحتملة للدرجات غير الصالحة. اطبع رسالة خطأ في حالة حدوث أي استثناء.

Input

```
Enter your grade (0-100): 105
```

Output

```
Error: Invalid grade. Please enter a grade between 0 and 100.
```

Solution

```
// www.gammal.tech

#include <iostream>

using namespace std;

int main() {
    int userGrade;

    cout << "Enter your grade (0-100): ";
    cin >> userGrade;

    try {
        if (userGrade < 0 || userGrade > 100) {
            throw "Invalid grade. Please enter a grade between 0 and 100.";
        }

        cout << "Your grade is: " << userGrade << endl;
    } catch (const char* errorMsg) {
        cerr << "Error: " << errorMsg << endl;
    }

    return 0;
}
```

6- Write a program that prompts the user to enter a number. If the entered number is less than or equal to 1000, throw an exception of type int. Catch the exception and print "Hi" if it occurs. Otherwise, print "Hello Gammal Tech".

اكتب برنامجًا يطلب من المستخدم إدخال رقم. إذا كان الرقم الذي تم إدخاله أقل من أو يساوي 1000، فقم بطرح استثناء من النوع int. اكتشف الاستثناء واطبع "Hi" في حالة حدوثه. بخلاف ذلك، قم بطباعة "Hello Gammal Tech".

Input

```
Enter a number: 500
```

Output

```
Hi
```

Solution

```
// www.gammal.tech

// Solution:
#include <iostream>

using namespace std;

int main() {
    try {
        int x;
        cout << "Enter a number: ";
        cin >> x;

        if (x <= 1000)
            throw 3; // If the condition is met, throw an exception of type int

        // If the condition is not met, this part will be executed
        cout << "Hello Gammal Tech" << endl;
    } catch (int x) {
        cout << "Hi" << endl; // Catch the exception and print "Hi"
    }

    return 0;
}
```

7- Write a program uses exception handling to catch different types of exceptions based on specific conditions. It prompts the user to enter a number and throws an exception of type int if the entered number is less than or equal to 1000, and an exception of type double if the number is greater than 1000 and less than or equal to 2000. It then catches and handles these exceptions accordingly.

كتابة برنامج يستخدم معالجة الاستثناءات لالتقاط أنواع مختلفة من الاستثناءات بناءً على شروط محددة. يطالب المستخدم بإدخال رقم ويطرح استثناءً من النوع int إذا كان الرقم المُدخل أقل من أو يساوي 1000، واستثناء من النوع double إذا كان الرقم أكبر من 1000 وأقل من أو يساوي 2000. ومن ثم يلتقط هذه الاستثناءات ويتعامل معها وفقاً لذلك.

Input

```
Enter a number: 1500
```

Output

```
Caught a double exception with value: 4.5
```

Solution

```
// www.gammal.tech

#include <iostream>

using namespace std;

int main() {
    try {
        int x;
        cout << "Enter a number: ";
        cin >> x;

        if (x <= 1000)
            throw 3; // If the condition is met, throw an exception of type int
        else if (x > 1000 && x <= 2000)
            throw 4.5; // If the condition is met, throw an exception of type double

        // If the conditions are not met, this part will be executed
        cout << "Hello Gammal Tech" << endl;
    } catch (int x) {
        cout << "Caught an int exception with value: " << x << endl; // Catch int exception
    } catch (double y) {
        cout << "Caught a double exception with value: " << y << endl; // Catch double exception
    }

    return 0;
}
```

8- Write a program that prompts the user to enter a number. If the entered number is between 1000 and 1500 (inclusive), throw an exception of type int. Catch the exception and print "Hi" if it occurs. Otherwise, print "Hello Gammal Tech".

اكتب برنامجًا يطلب من المستخدم إدخال رقم. إذا كان الرقم الذي تم إدخاله يتراوح بين 1000 و1500 (ضمنياً)، فقم بطرح استثناء من النوع int. اكتشف الاستثناء واطبع "Hi" في حالة حدوثه. بخلاف ذلك، قم بطباعة "Hello Gammal Tech".

Input

Enter a number: 1400

Output

Hi

Solution

```
// www.gammal.tech

// Solution:
#include <iostream>

using namespace std;

int main() {
    try {
        int x;
        cout << "Enter a number: ";
        cin >> x;

        if (x >= 1000 && x <= 1500)
            throw 3; // If the condition is met, throw an exception of type int

        // If the conditions are not met, this part will be executed
        cout << "Hello Gammal Tech" << endl;
    } catch (int x) {
        cout << "Hi" << endl; // Catch the exception and print "Hi"
    }

    return 0;
}
```


9- Write a program that prompts the user to enter a number. If the entered number is between 1000 and 1500 (inclusive), throw an exception of type int. Catch the exception and print "x" if it occurs. Otherwise, print "HI".

اكتب برنامجًا يطلب من المستخدم إدخال رقم. إذا كان الرقم الذي تم إدخاله يتراوح بين 1000 و1500 (ضمنًا)، فقم بطرح استثناء من النوع int. احصل على الاستثناء واطبع "x" في حالة حدوثه. بخلاف ذلك، قم بطباعة "مرحبًا".

Input

```
Enter a number: 15
```

Output

```
HI
```

Solution

```
// www.gammal.tech
// Solution:
#include <iostream>

using namespace std;

int main() {
    try {
        int x;
        cout << "Enter a number: ";
        cin >> x;

        if (x >= 1000 && x <= 1500)
            throw 3;

        cout << "HI" << endl;
    } catch (int x) {
        cout << "x" << endl;
    }

    return 0;
}
```

10- Write a program that prompts the user to enter a number. If the entered number is both between 1000 and 1500 (inclusive) and even, throw an exception of type int. Catch the exception and print "x" if it occurs. Otherwise, print "HI".

اكتب برنامجًا يطلب من المستخدم إدخال رقم. إذا كان الرقم الذي تم إدخاله يتراوح بين 1000 و1500 (ضمنًا) وحتى، فقم بطرح استثناء من النوع int. احصل على الاستثناء واطبع "x" في حالة حدوثه. بخلاف ذلك، قم بطباعة "مرحبًا".

Input

Enter a number: 1002

Output

x

Solution

```
// www.gammal.tech
// Solution:
#include <iostream>

using namespace std;

int main() {
    try {
        int x;
        cout << "Enter a number: ";
        cin >> x;

        if (x >= 1000 && x <= 1500 && x % 2 == 0)
            throw 3; // If the conditions are met, throw an exception of type int

        // If the conditions are not met, this part will be executed
        cout << "HI" << endl;
    } catch (int x) {
        cout << "x" << endl; // Catch the exception and print "x"
    }

    return 0;
}
```