# Lab 7.2.1 Exceptions: input validation

## Objectives

Familiarize the student with:

- identifying situations when exceptions should be thrown;
- adding code to handle exceptions;
- adding code to throw exceptions.

## Scenario

Try to change the program below (both the div and main functions). Insert a throw and a try-catch block instead of a Boolean value and res parameter (change it to a function return value). Remember to inform the user of any exceptions in simple words. Try to make two versions of your code, one that catches the division by zero exception, and another that checks arguments and then throws an exception (when the argument isn't proper). Which version is better?

```
#include <iostream>

using namespace std;
const int DivideByZero = 111;
using namespace std;
float internaldiv(float arg1, float arg2)
{
 if (0==arg2)
  throw DivideByZero;
 return arg1 / arg2;
}

bool div(float &res, float arg1, float arg2) {
 if(arg2 == 0.0)
  return false;
 internaldiv(arg1, arg2);
 return true;
}

int main(void) {
 float r, a, b;
 while(cin >> a) {
  cin >> b;
  if(div(r,a,b))
   cout << r << endl;
  else
   cout << "Are you kidding me?" << endl;

 }
 return 0;
}
```

### Example input

```
2
1
```

### Example output

```
2
```

## Example input

```
6
0
```

## Example output

```
Are you kidding me?
Your input is not valid. You can't divide by zero.
```