Lab 5.3.5 Modelling fractions: part 1

Objectives

Familiarize the student with:

- modelling real-world entities with classes and objects;
- providing meaningful and helpful representations of objects.

Scenario

We've already seen that floating-point numbers have their quirks in C++.

Let's try to write our own class representing non-integer numbers, namely fractions!

Our class shall, for now, provide the following functionalities:

- the class will provide a textual representation;
- the textual representation willdisplay whole parts correctly, i.e. 7/4 will be displayed as "1 3/4";
- the textual representation will display signs correctly, i.e. -3/4 and 3/-4 will be displayed as "-3/4";
- the class willprovide a floating-point representation;
- the textual representation will display whole parts correctly, i.e. 7/4 will be displayed as "1 3/4";
- We'll assume the denominator will not be set to 0.

Your program should allow input in the form "[numerator] / [denominator]".

```
#include <iostream>
#include <string>

using namespace std;

class Fraction{
public:
  Fraction(int numerator, int denominator);
  string toString();
  double toDouble();
private:
  int numerator;
  int denominator;
};

// implement Fraction methods

int main(void) {
  int num, den;
  std::string input = "";
  std::cin.getline(input);

  // parse input and get numerator and denominator

  Fraction fraction(num, den);

 cout << fraction.toString() << " is " fraction.toDouble() in decimal << endl;
 return 0;
}
```

Example input

```
3 / 4
```

Example output

```
3/4 is 0.75 in decimal
```

Example input

```
7 / -4
```

Example output

```
-1 3/4 is -1.75 in decimal
```

Example input

```
8 / 4
```

Example output

```
2 is 2.0 in decimal
```

Example input

```
-6 / 8
```

Example output

```
-6/8 is -0.75 in decimal
```