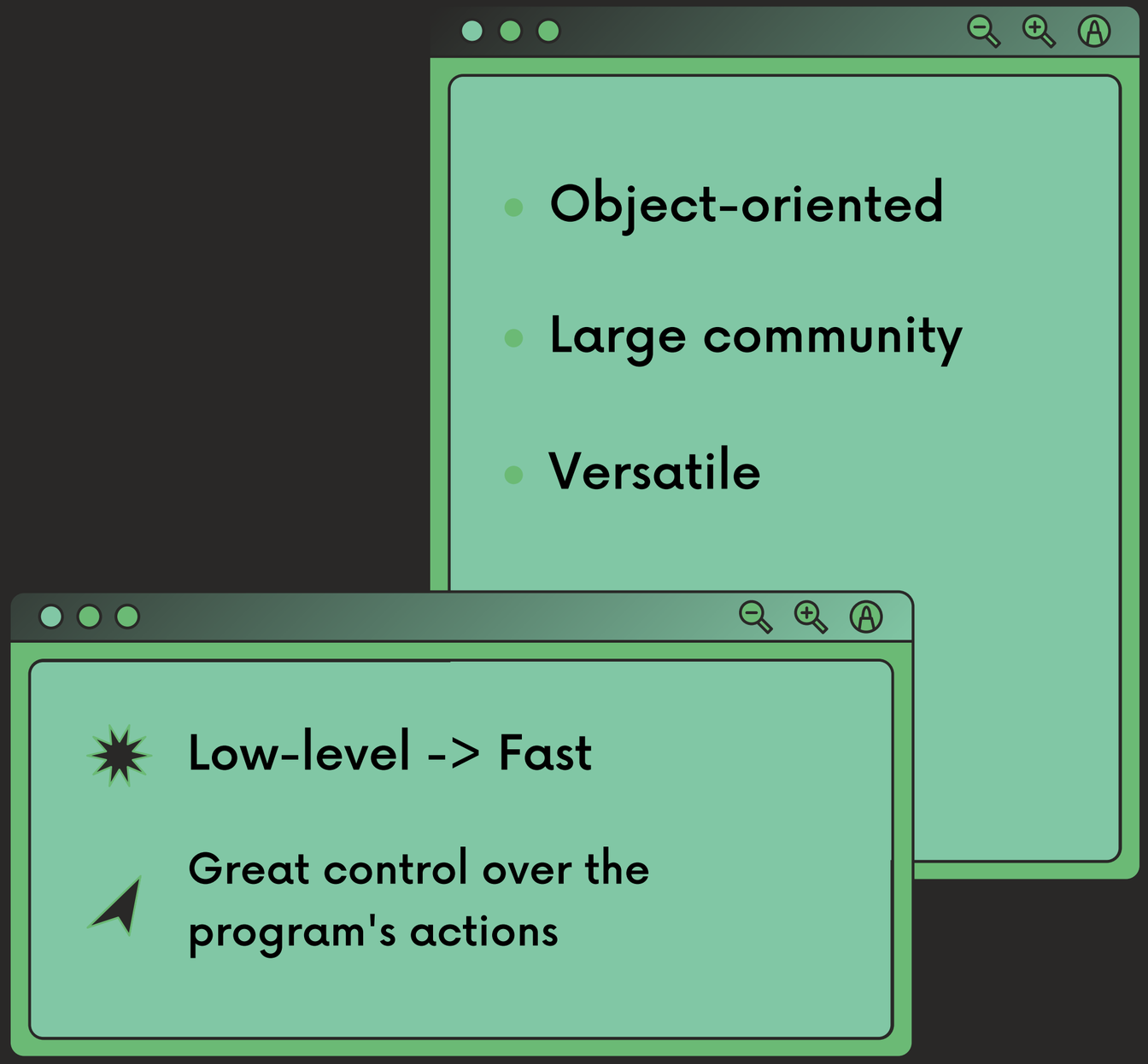
# C++ Basics

# Why C++?


Object-oriented
Large community
Versatile
Low-level -> Fast
Great control over the program's actions

# My First Program in C++

# Namespace std

The namespace function is introduced so we're able to give the same name to multiple variables.
Namespace std is a common namespace, which includes many useful objects for C++, like vectors, etc.

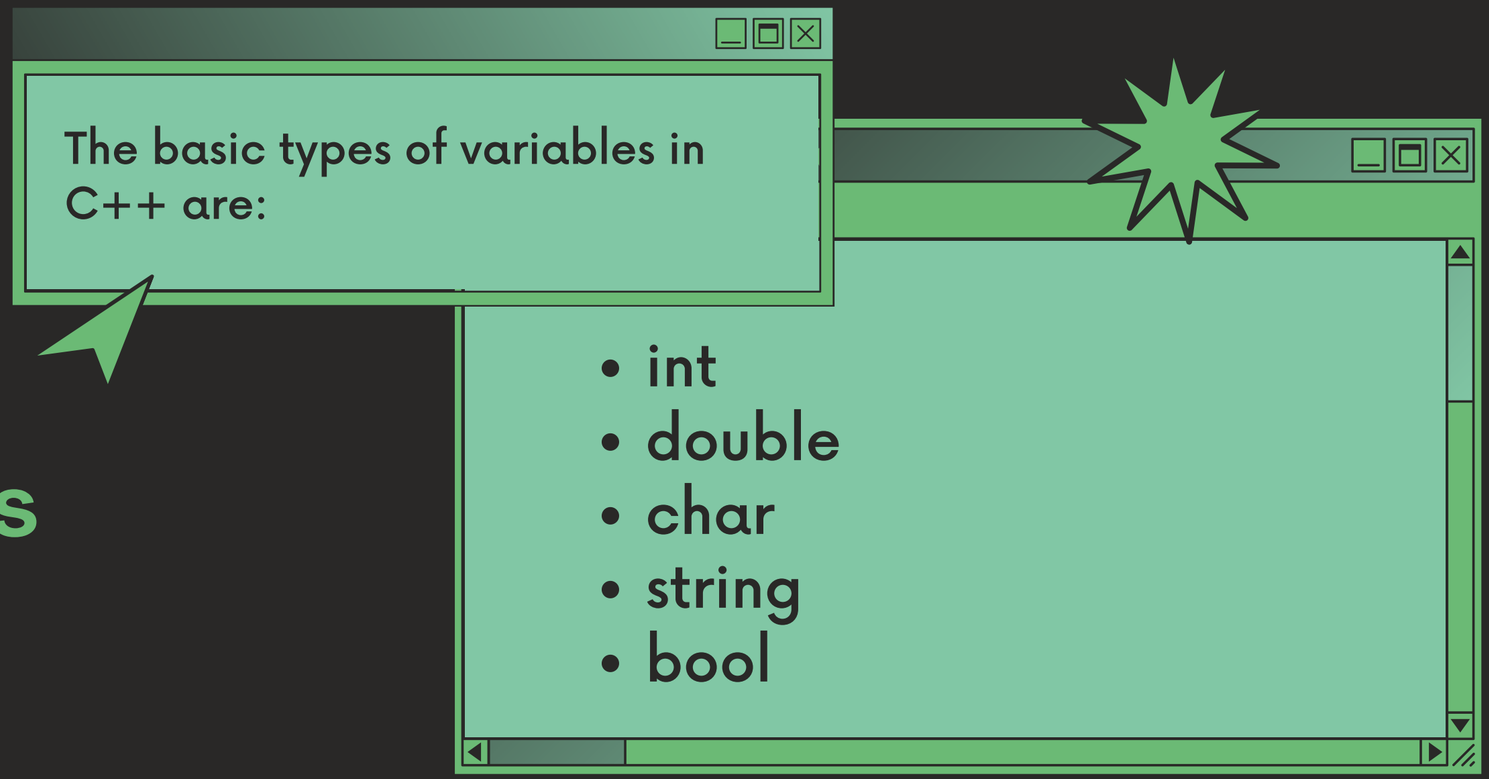
# C++ is a compiled Language

C++ is usually compiled using g++.

Check if you have g++ installed by running "g++ --version".

Try compiling your first program by running g++ -o name.exe name.cpp!

# Variable Types




But you can get many others though packages!

# Variable Declaration

- Unlike other higher-level languages, in C++ you need to indicate the variable type when declaring a variable.
- The naming convention is camelCase, but...

# Input/Output

You can output and input to the terminal easily in C++.

```
#include <iostream>
```

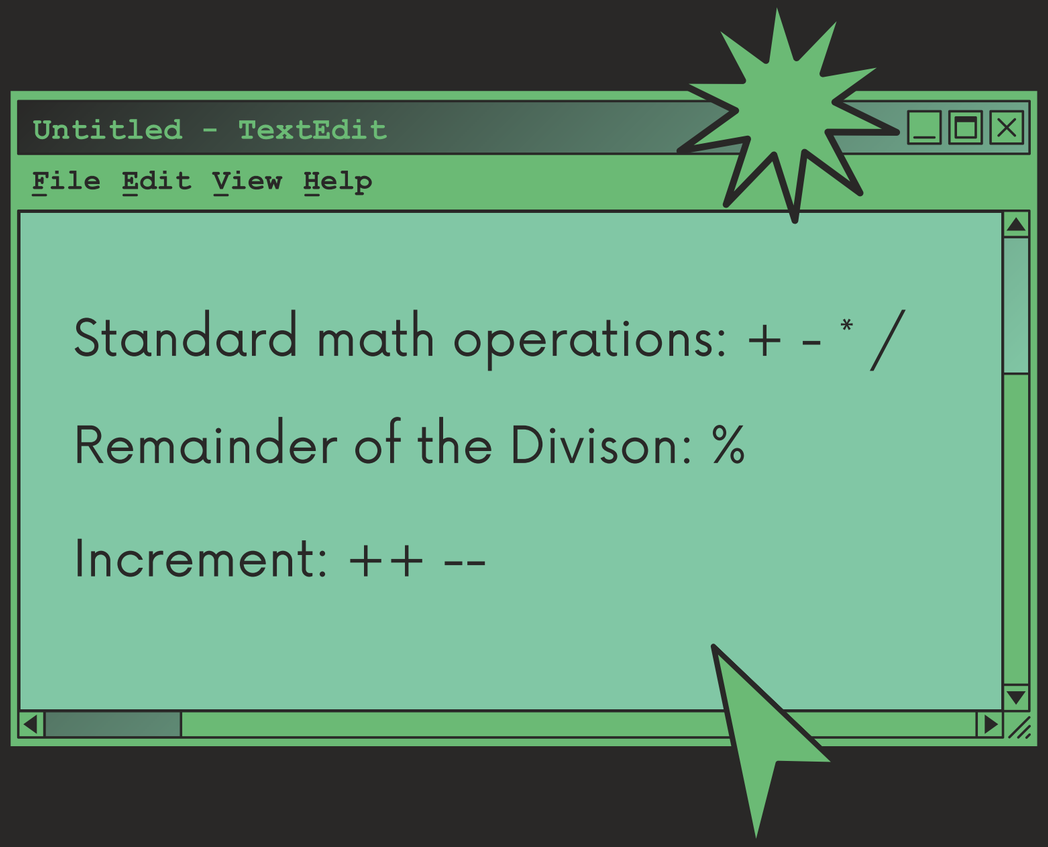
# Input/Output from/to Files

You can output to files pretty much the same as to the terminal.

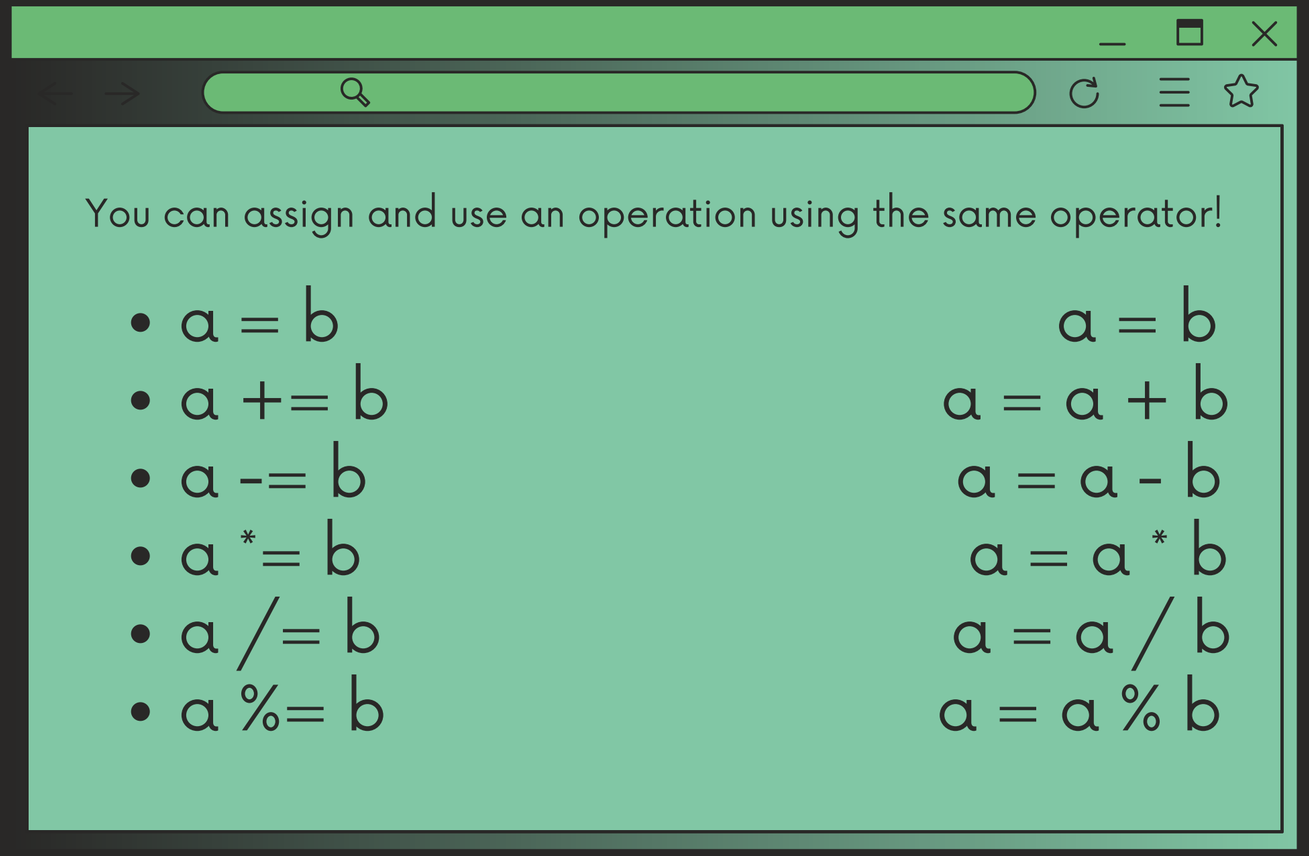
You can also get data from files (the easiest way being line by line)

# Arithmetic Operators

C++ is famous particularly due to its mathematical capacities. Here are their math operators!



# Assignment Operators

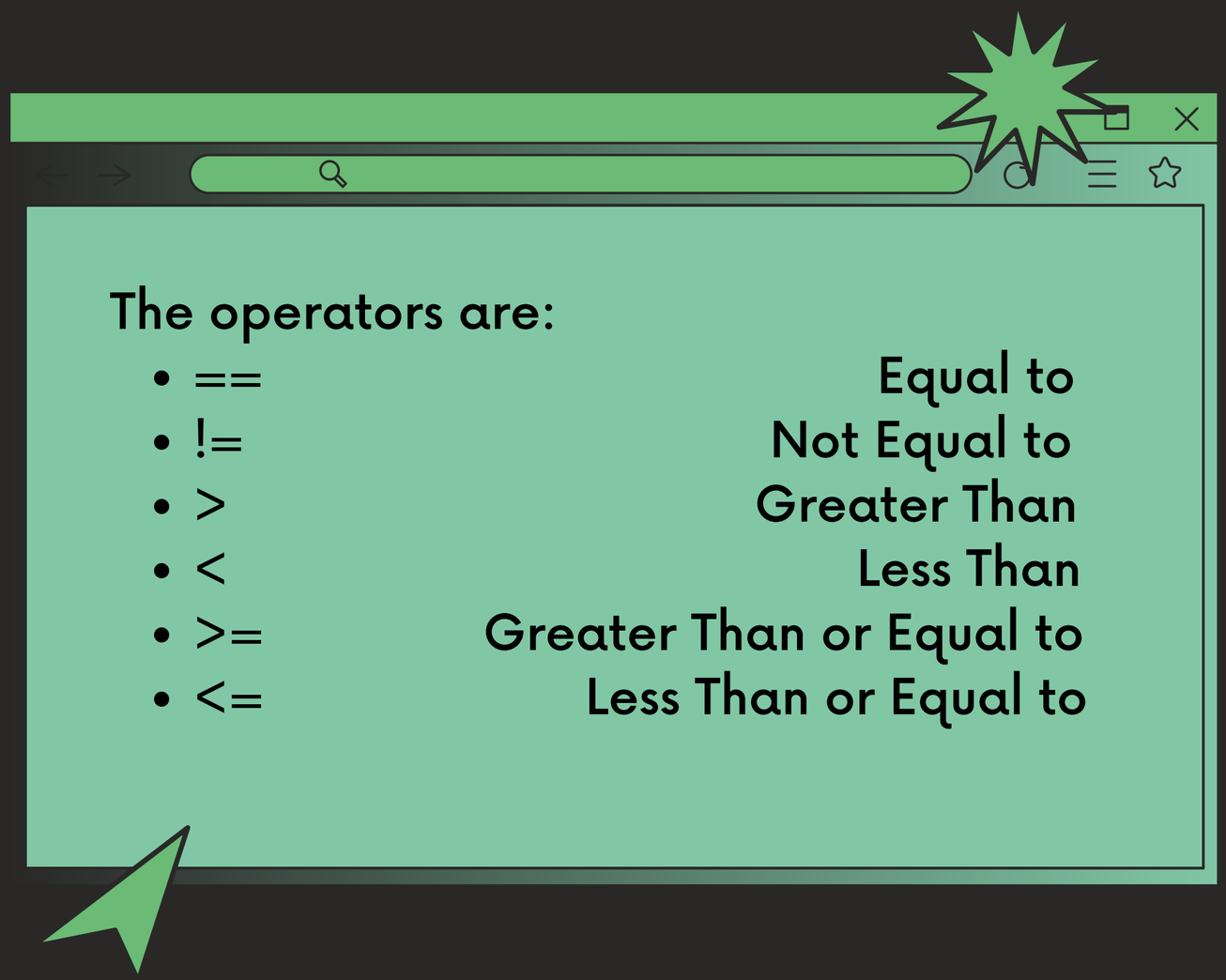


You can assign and use an operation using the same operator!

| | |
|---|---|
| • a = b | a = b |
| • a += b | a = a + b |
| • a -= b | a = a - b |
| • a *= b | a = a * b |
| • a /= b | a = a / b |
| • a %= b | a = a % b |

# Relational Operators

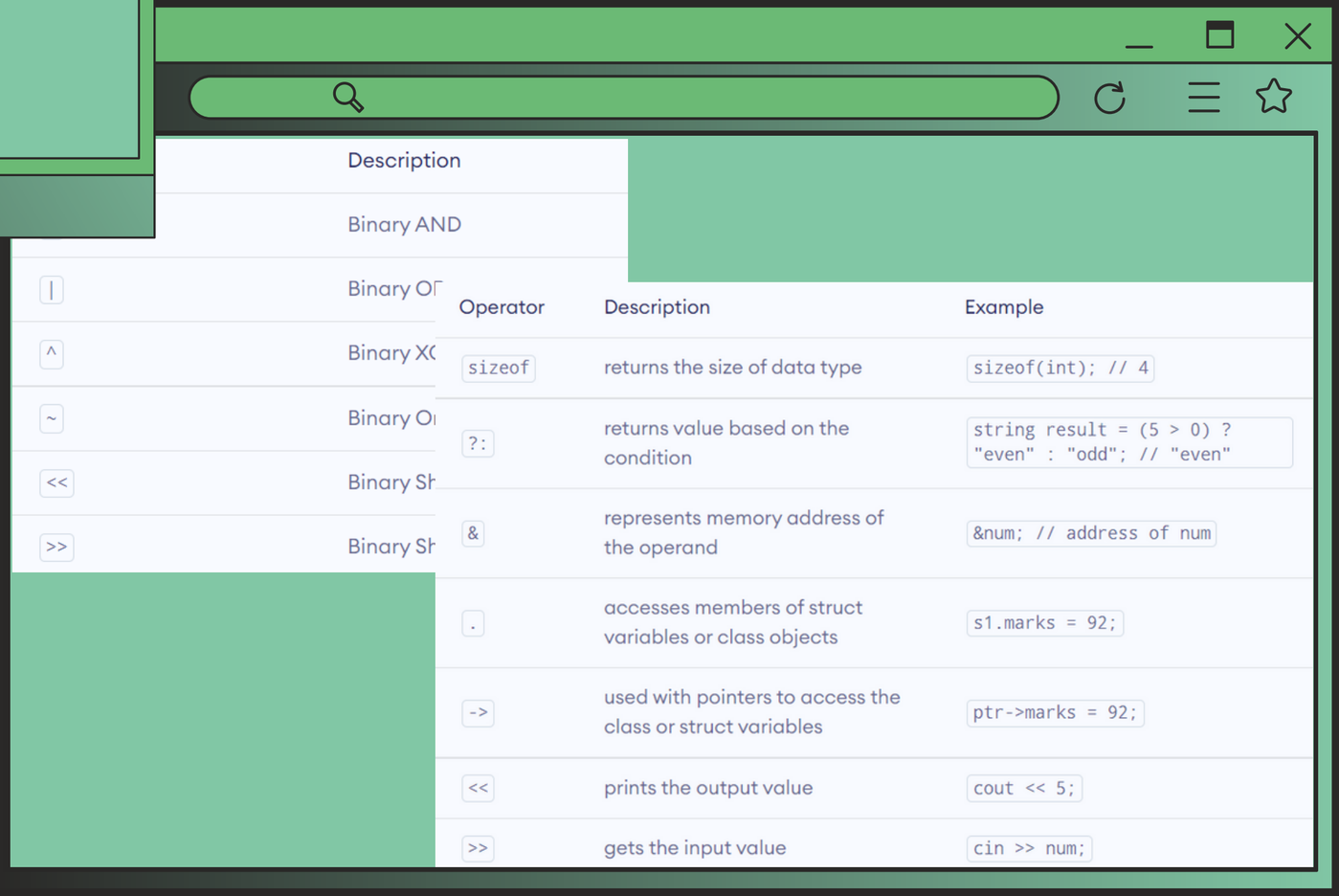
- You can compare two statements and get a boolean back.



# Logical Operators (& Many More)

To complete your expressions, you still have the following operators:

- && AND
- || OR
- ! NOT



| Description |
| --- |
| Binary AND |
| Binary OR |
| Binary XO |
| Binary O |
| Binary Sh |
| Binary Sh |

| Operator | Description | Example |
| --- | --- | --- |
| `sizeof` | returns the size of data type | `sizeof(int); // 4` |
| `?:` | returns value based on the condition | `string result = (5 > 0) ? "even" : "odd"; // "even"` |
| `&` | represents memory address of the operand | `&num; // address of num` |
| `.` | accesses members of struct variables or class objects | `s1.marks = 92;` |
| `->` | used with pointers to access the class or struct variables | `ptr->marks = 92;` |
| `<<` | prints the output value | `cout << 5;` |
| `>>` | gets the input value | `cin >> num;` |

# If Statements

The If Statements check a condition and run a block of code if the condition is true.

Use Else If when checking for multiple exclusive conditions and use Else to run code when none of the conditions is true.

You can also use Switch to check for the value of a variable.

Also useful: the ternary operator!

# Loops

C++ has many loop types, each useful for a specific thing:

- For
- Foreach
- While
- Do ... While
- Nested loops!

# Functions

Functions are a way to keep your code clean and to not have to repeat the same thing many times.

Simple lines like printf() like we've been using are functions, but you can write your own!

A function starts with the return type, then the name and then the parameters in parenthesis.

# Object-Oriented Programming

## What are classes and what are objects?

A class is like a blank form to fill out.
An object is like one of the responses to the form!

# Classes

A class in C++ is made up of its identifier (the word "class"), the name you want the give the class (the name of your "form") and then you can specifiy the variables of your class (these are the questions on your "form") and if they're public or private.

# Objects: Instances of Classes

Like we mentioned before, an object is one of the answers to your "form".

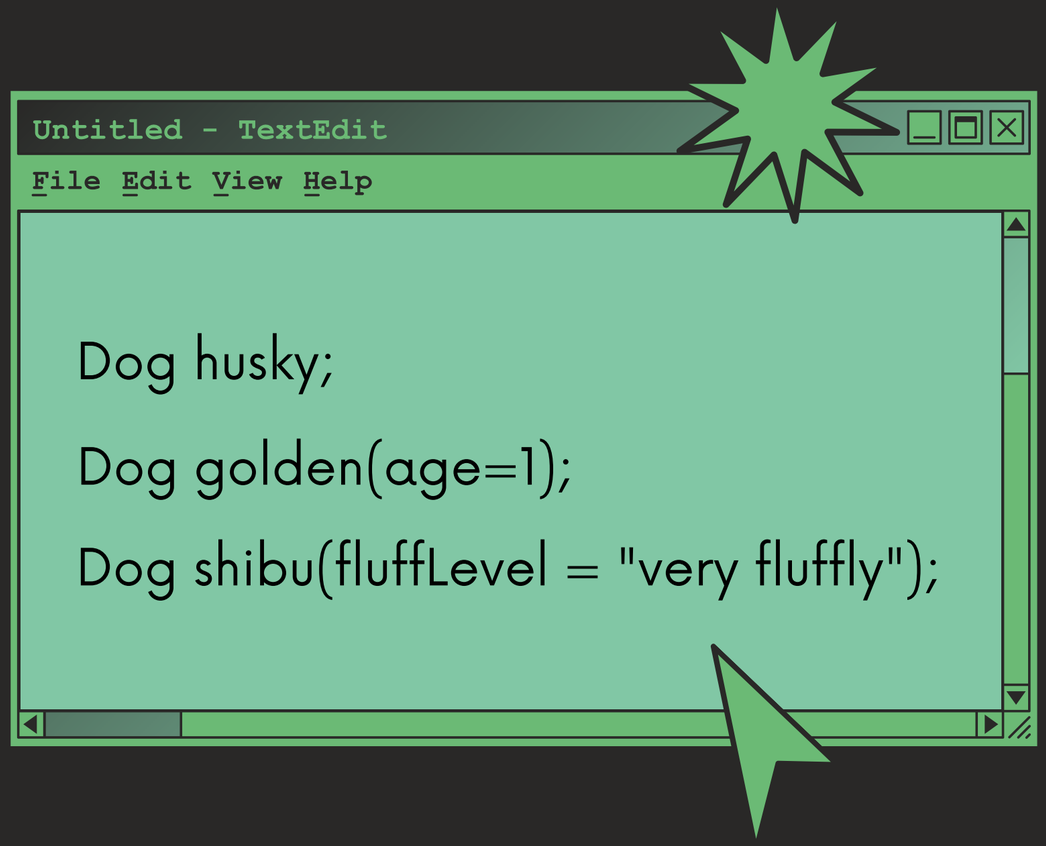
In C++, you create an object by calling for the name of your class and then giving it a name, the same way you declare a variable.

# Class Methods

- A method is a function that belongs to a class.
- You create it the same way you would a normal function, but inside the class environment, and you call it by using object.method().

# The Constructor Method

Like the name suggests, the constructor method is the method that actually constructs your objects. In C++, you can create an object in many ways!



# Bonus: Encapsulation

When creating a class, you can choose to leave your variables public or private.

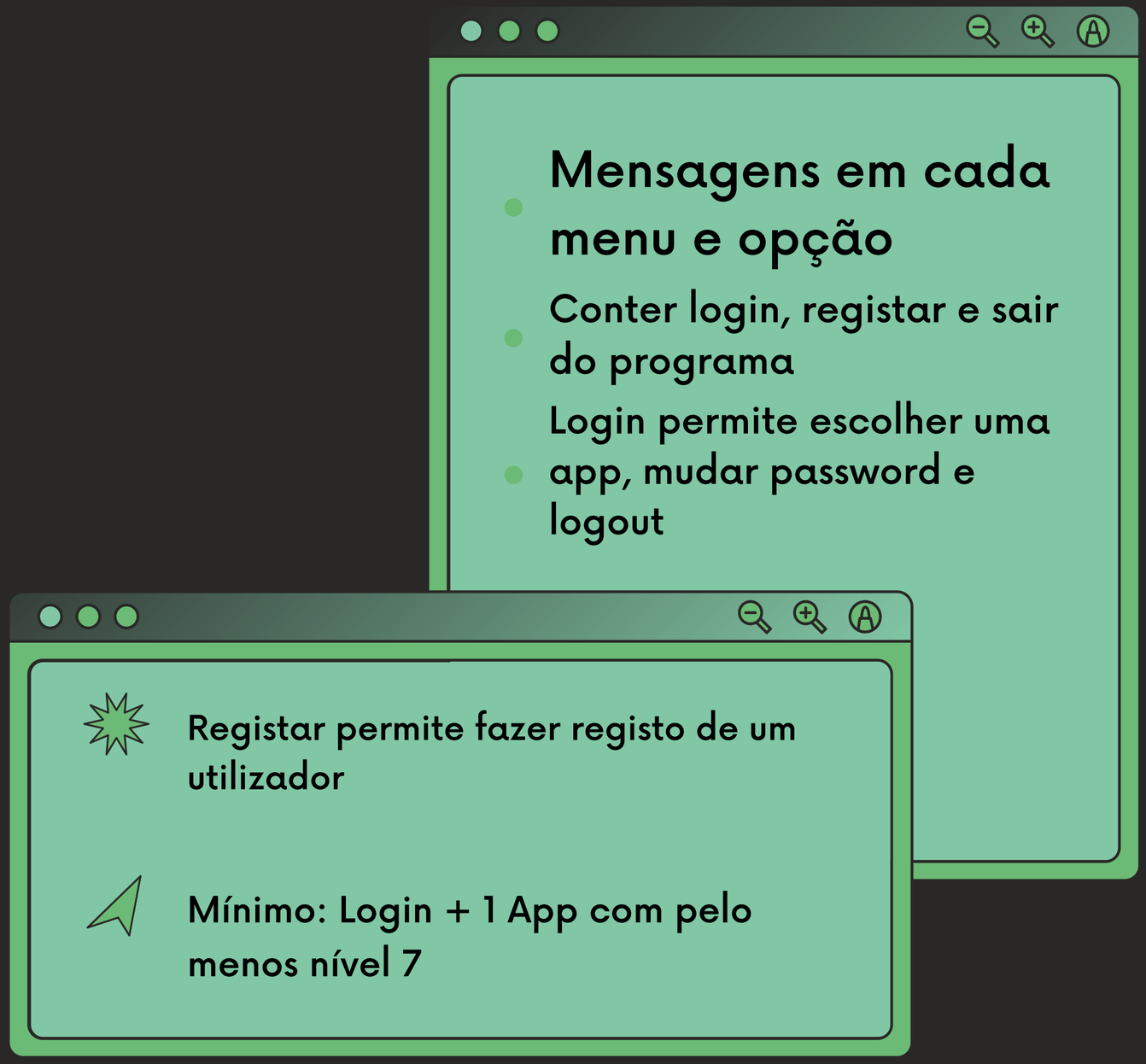
It is considered good practice to keep your variables private and access them when need be using get methods.

# Bonus: Inheritance

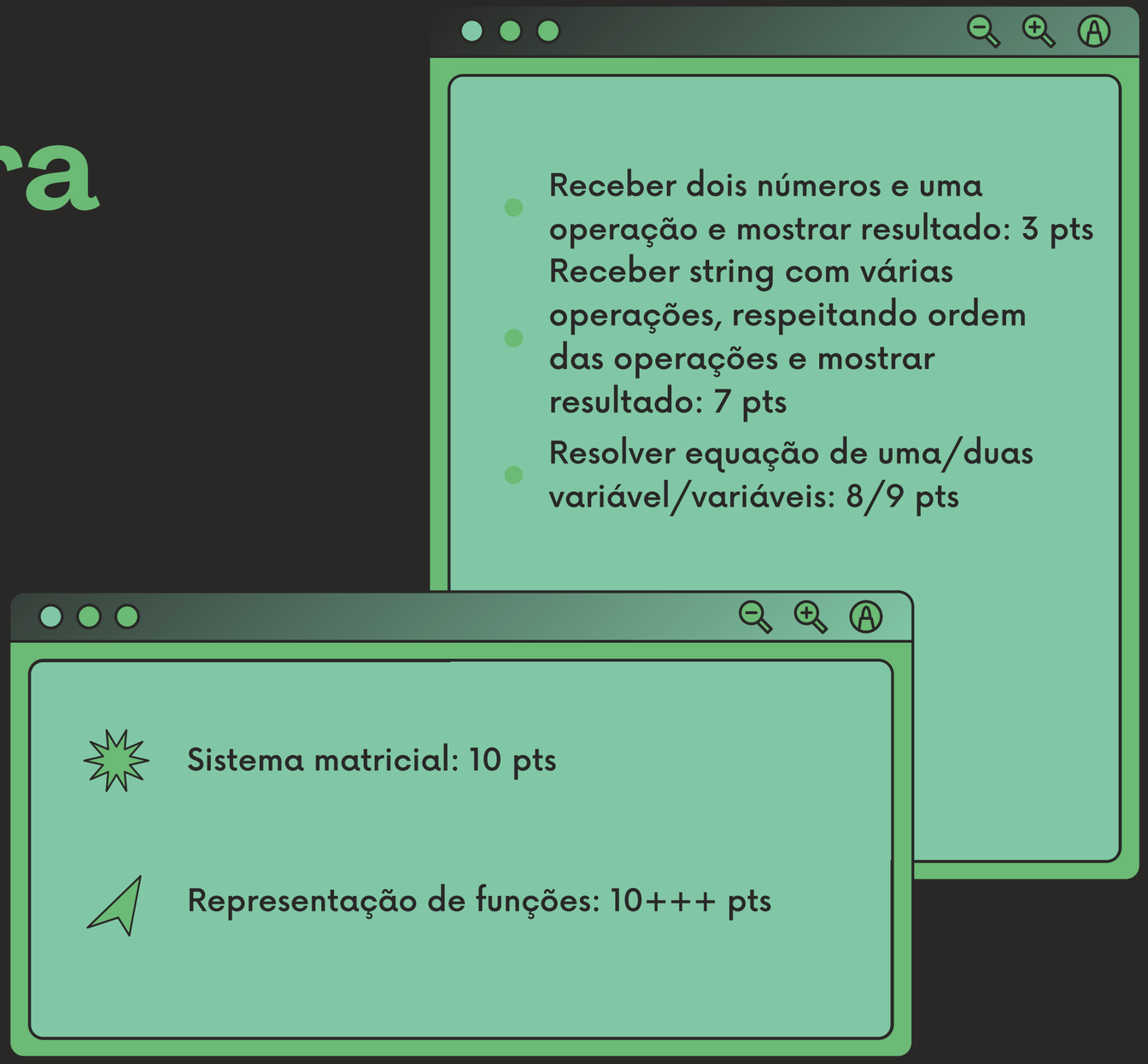
- Sometimes it makes sense to build a new class and have that class access attributes and methodsmfrom another class.
- For that, we create child/parent classes.

# Projeto Final

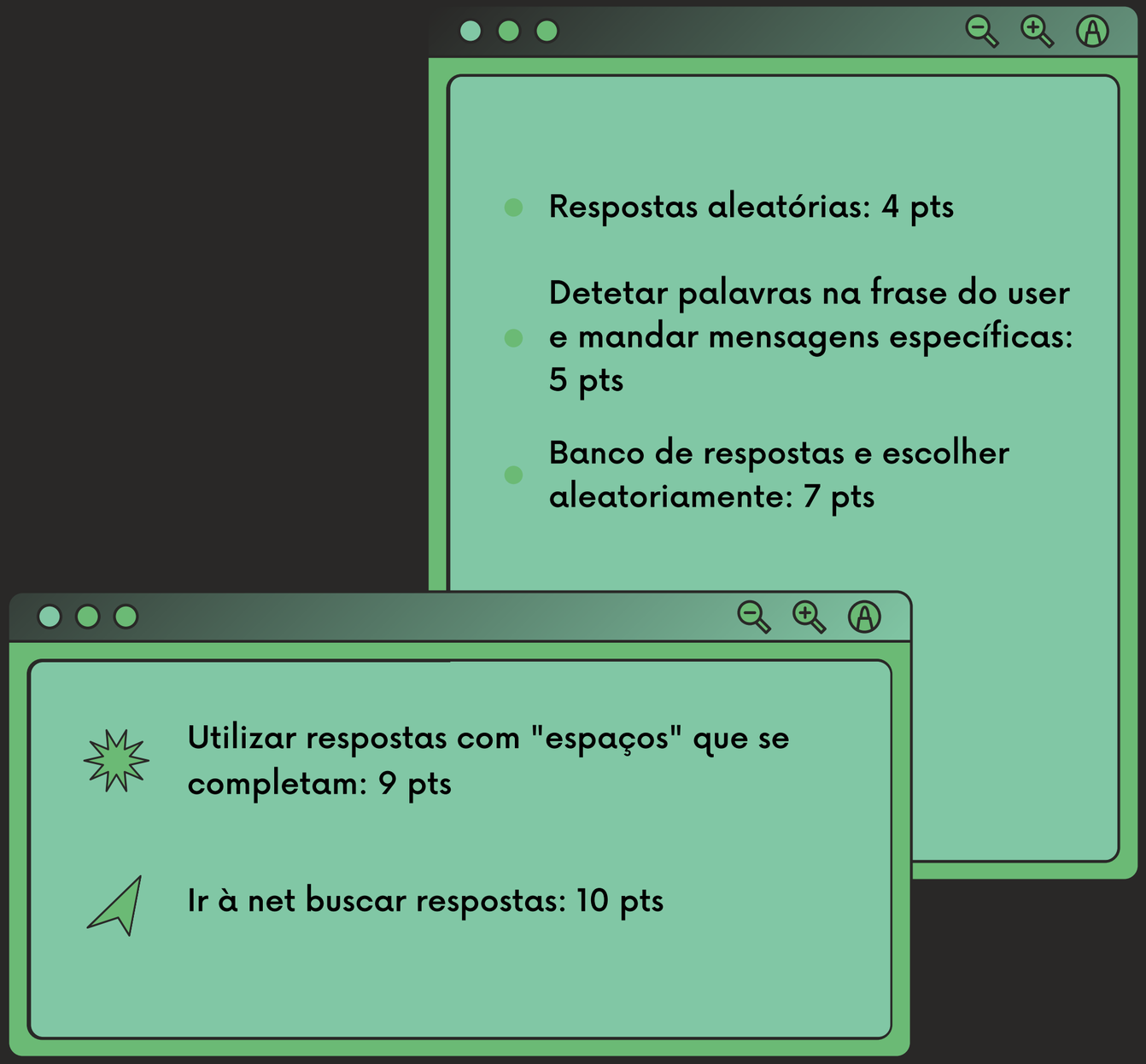
# Login com Apps



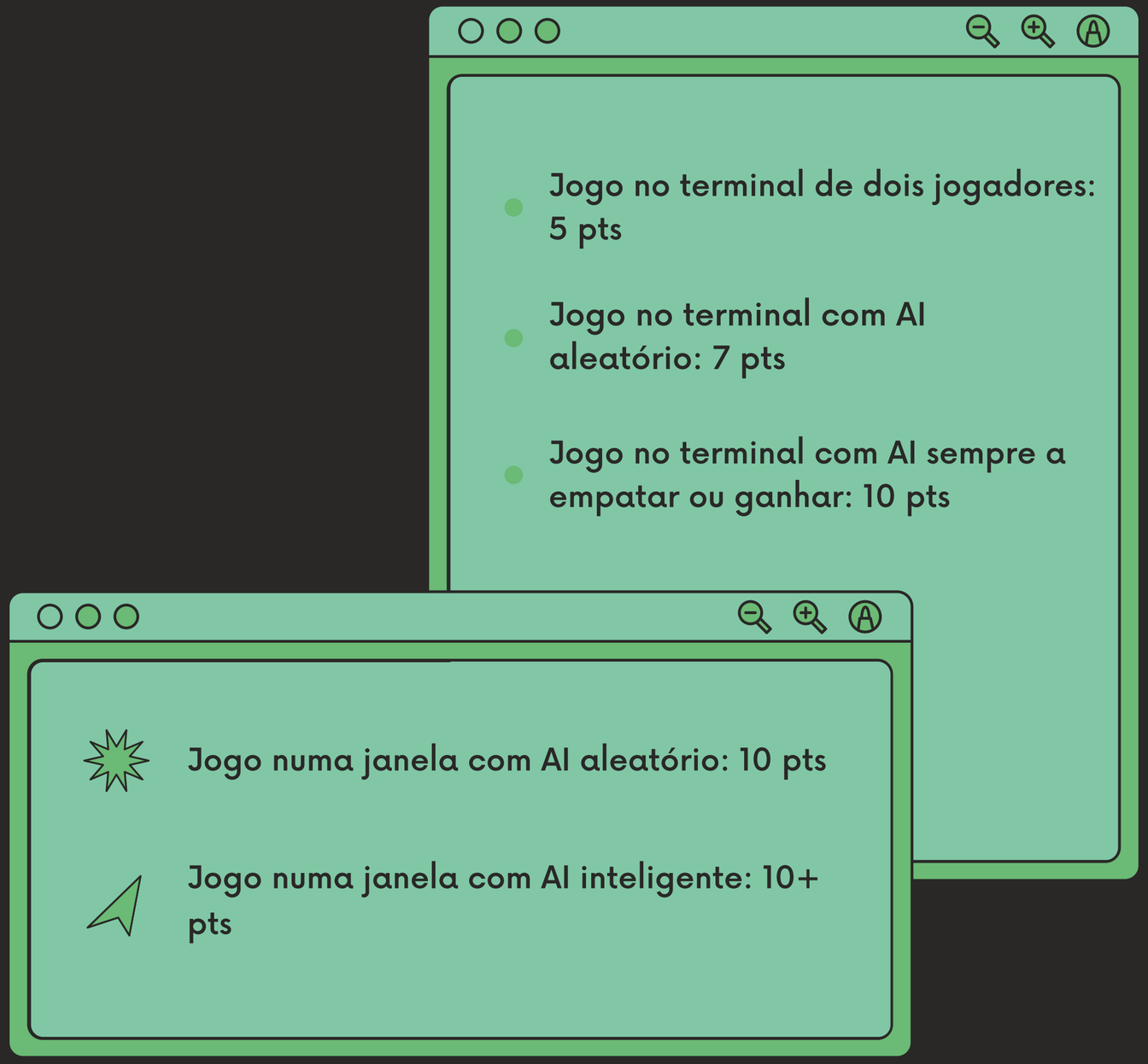
# Calculadora



# ChatBot


Respostas aleatórias: 4 pts
Detetar palavras na frase do user e mandar mensagens específicas: 5 pts
Banco de respostas e escolher aleatoriamente: 7 pts
Utilizar respostas com "espaços" que se completam: 9 pts
Ir à net buscar respostas: 10 pts

# TicTacToe


Jogo no terminal de dois jogadores: 5 pts
Jogo no terminal com AI aleatório: 7 pts
Jogo no terminal com AI sempre a empatar ou ganhar: 10 pts
Jogo numa janela com AI aleatório: 10 pts
Jogo numa janela com AI inteligente: 10+ pts

# Obrigada pela atenção!

Boa sorte para o projeto! Qualquer dúvida, falem comigo.

**Referências:**


Francisco Fonseca Wiki

w3schools.com

cplusplus