

What is your name?

danil

shahzoda malikova

bahromboy

abduraxmon

abdulaziz makhmudov

leonid voronin

fazogir

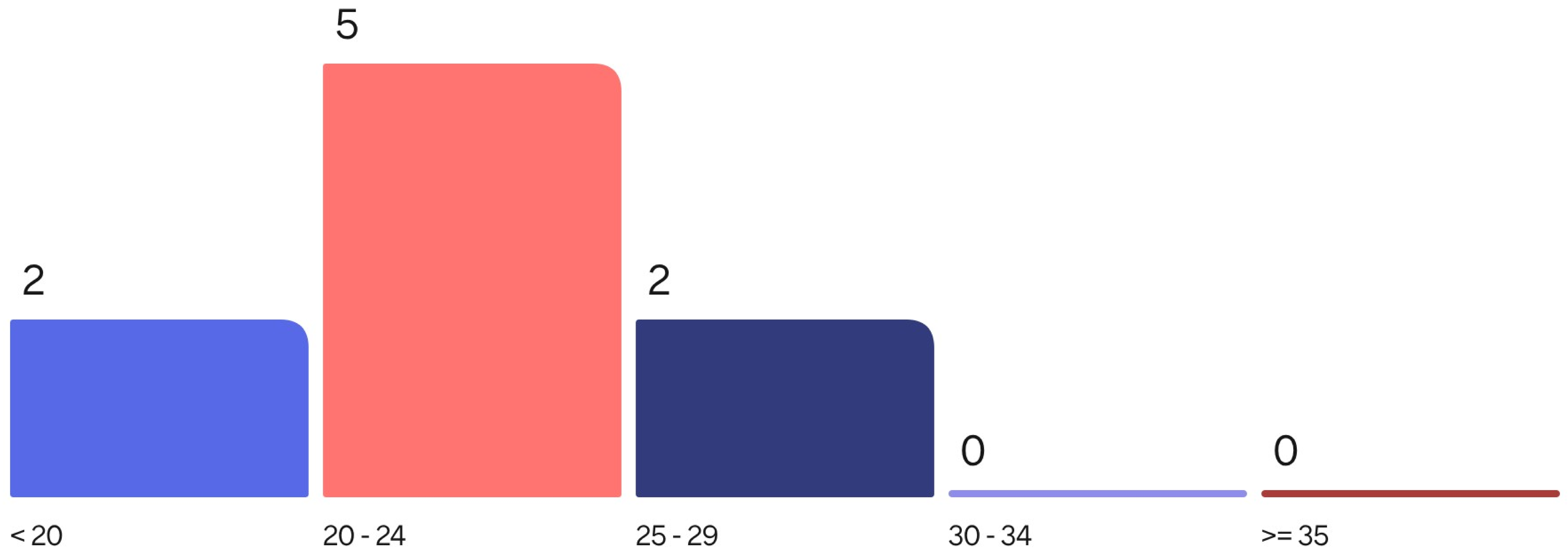
rafkat gatin

nurali ahmatov

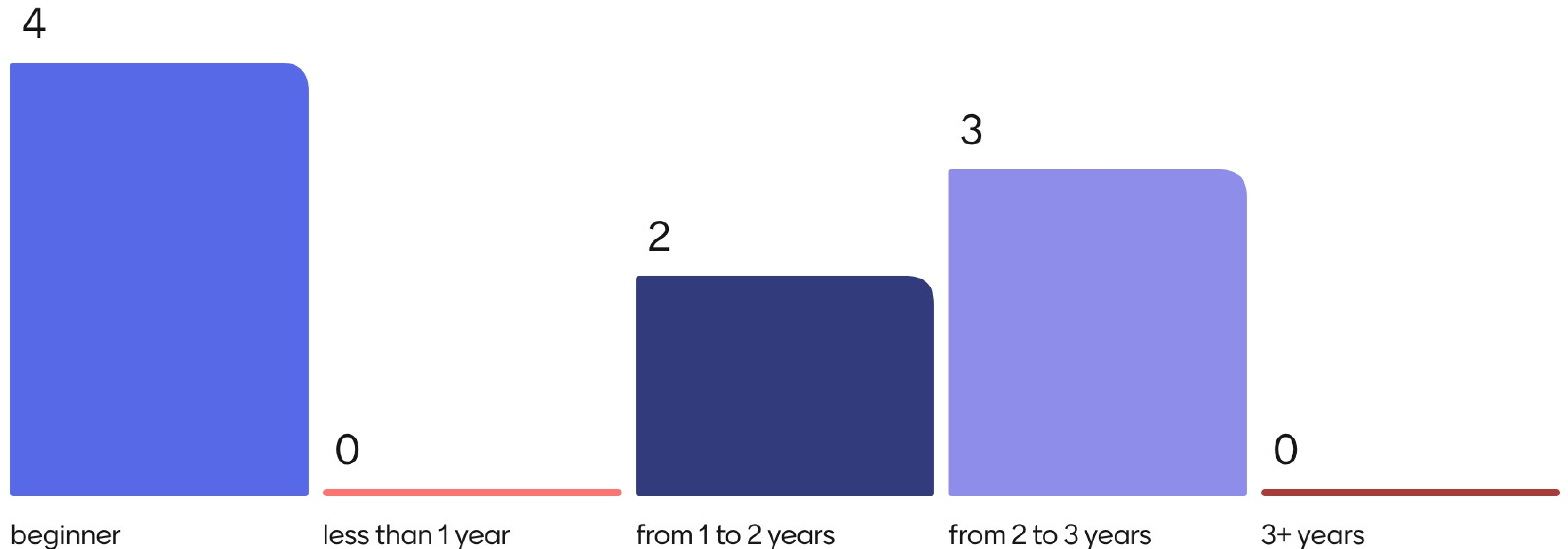
What city (or region) are you from?

tashkent
fergana urgench

How old are you?



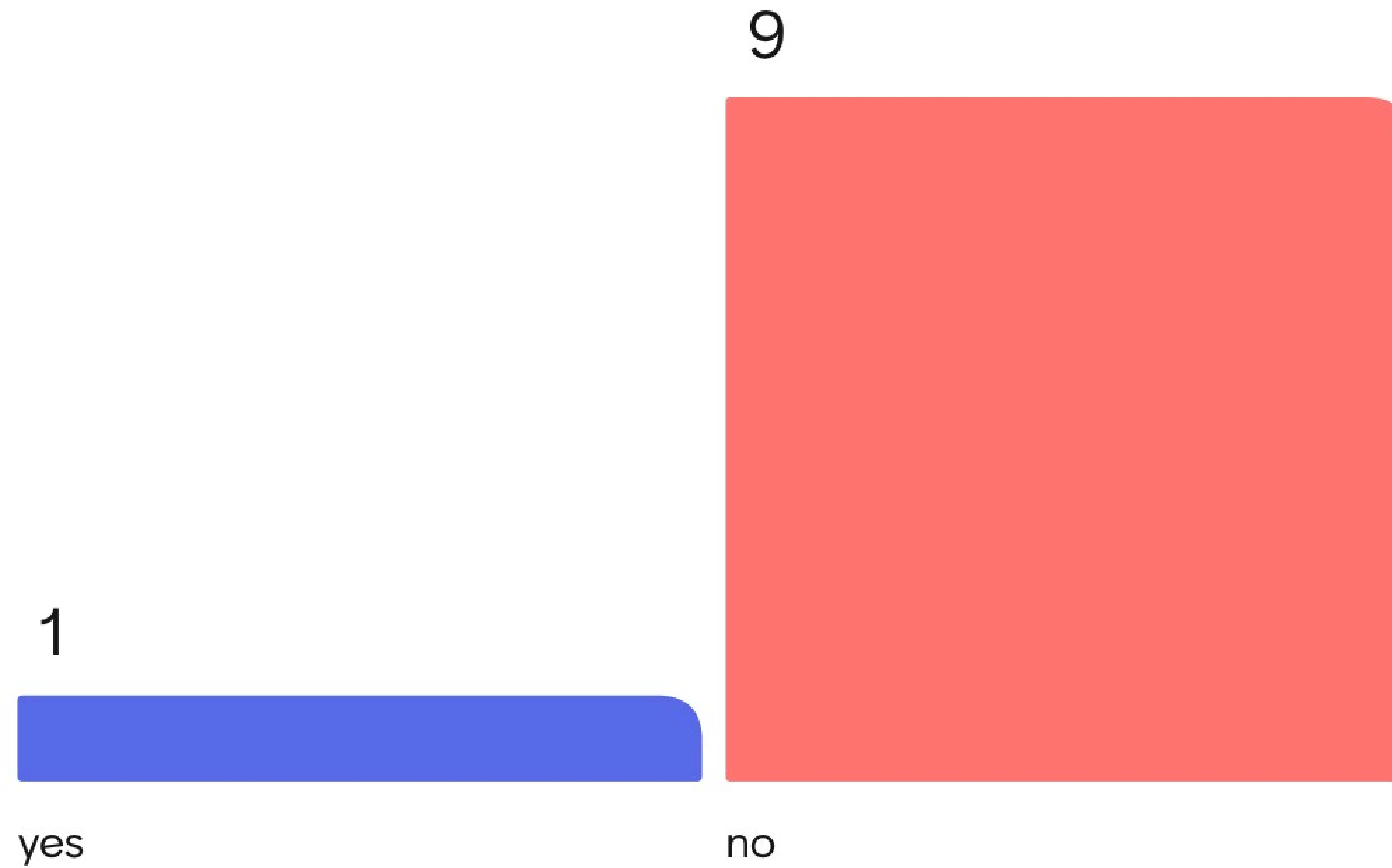
How long have you been programming?



What programming languages have you picked up??

javascript
js ts php
typescript
java js c
javascript

Have you looked into the first topic?



OOD and OOP stand for ...

Object oriented design
and Object oriented
programming

Object-Oriented
Programming Object-
Oriented Design

Object-oriented Design
and Object-oriented
Patterns

object oriented
design/programming

OOD stands for Object-Oriented Design, which is the blueprint or **planning phase** for structuring software using object-oriented principles, defining objects, their properties, behaviors, and relationships.

OOP stands for Object-Oriented Programming, which is the **implementation phase** where the OOD is translated into actual code using an object-oriented programming language to create classes and objects.

OOD (Object-Oriented Design)

Purpose:

To create a high-level conceptual model and architecture for a software system.

Activities:

Identifying objects, their attributes (data), methods (behaviors), and the interactions and relationships between them.

Nature:

Abstract and conceptual, providing a blueprint before coding begins.

Outcome:

A well-defined structure for the system that emphasizes maintainability and scalability.

OOP (Object-Oriented Programming)

Purpose:

To implement the design into a working software using a programming language.

Activities:

Writing code, creating classes based on the design, instantiating objects, and implementing the methods to define object behaviors.

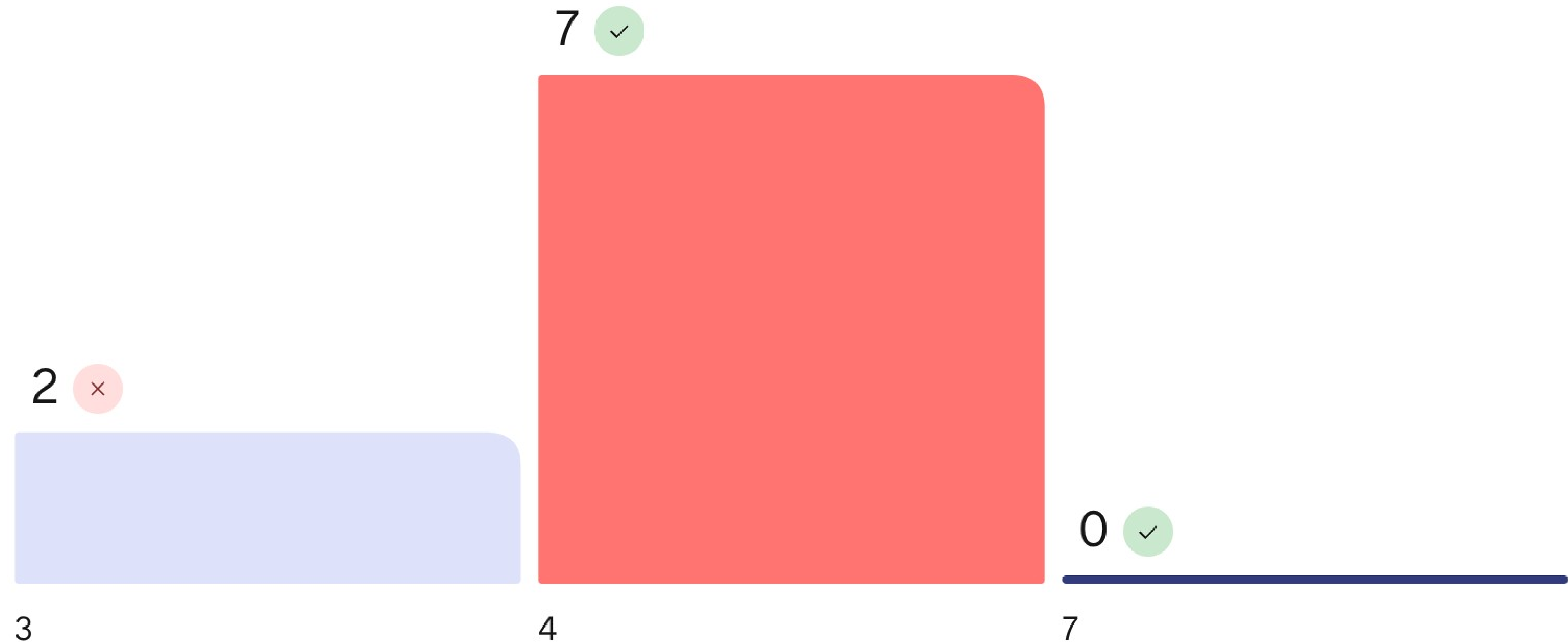
Nature:

Practical and hands-on, translating the design into executable code.

Outcome:

A functional software program built

How many principles is there in OOP?



List OOP Principles

Encapsulation,
Abstraction

inheritance,
incapsulation,
abstraction,
polymorphism

Encapsulation,
Abstraction, Inheritance,
Polymorphism

Encapsulation/
Abstraction/
Polymorphism/
Inheritance

encapsulation,
inheritance, abstraction,
polymorphism

Encapsulation,
Abstraction,
Polymorphism,
Inheritance.

Provide examples of Polymorphism

no idea

same

<-- same

Provide examples of Abstraction

public class Player which just describe method Play() without implementation, and class inheritor VideoPlayer which implement play function

Class Figure that has abstract method draw() . And Subclasses Circle, Square that implement this method differently

Provide examples of Inheritance

Class Car with some basic fields and Class ElectricalCar that extends Car class and add new field BatteryCapacity

Provide examples of Incapsulation

Highlight another famous acronym used in OOP?

SOLID and Grasp

SRP, OCP, LSP, ISP, DIP -
SOLID

SOLID. S – Single Responsibility Principle (SRP). O – Open/Closed Principle (OCP). L – Liskov Substitution Principle (LSP). I – Interface Segregation Principle (ISP). D – Dependency Inversion

SOLID

Principles

- Single responsibility
- Open–closed
- Liskov substitution
- Interface segregation
- Dependency inversion

Few more

- Encapsulate what varies
- Favor composition, not inheritance
- Program to interface
- Loose coupling

Design Pattern vs Design Principle

Principle is like general guidelines and Pattern is specific solution for a specific problem

Design Principles are high-level guidelines for good software design. They are the "why" behind the design. Design Patterns are specific, proven solutions to common problems. They are the "how" to im

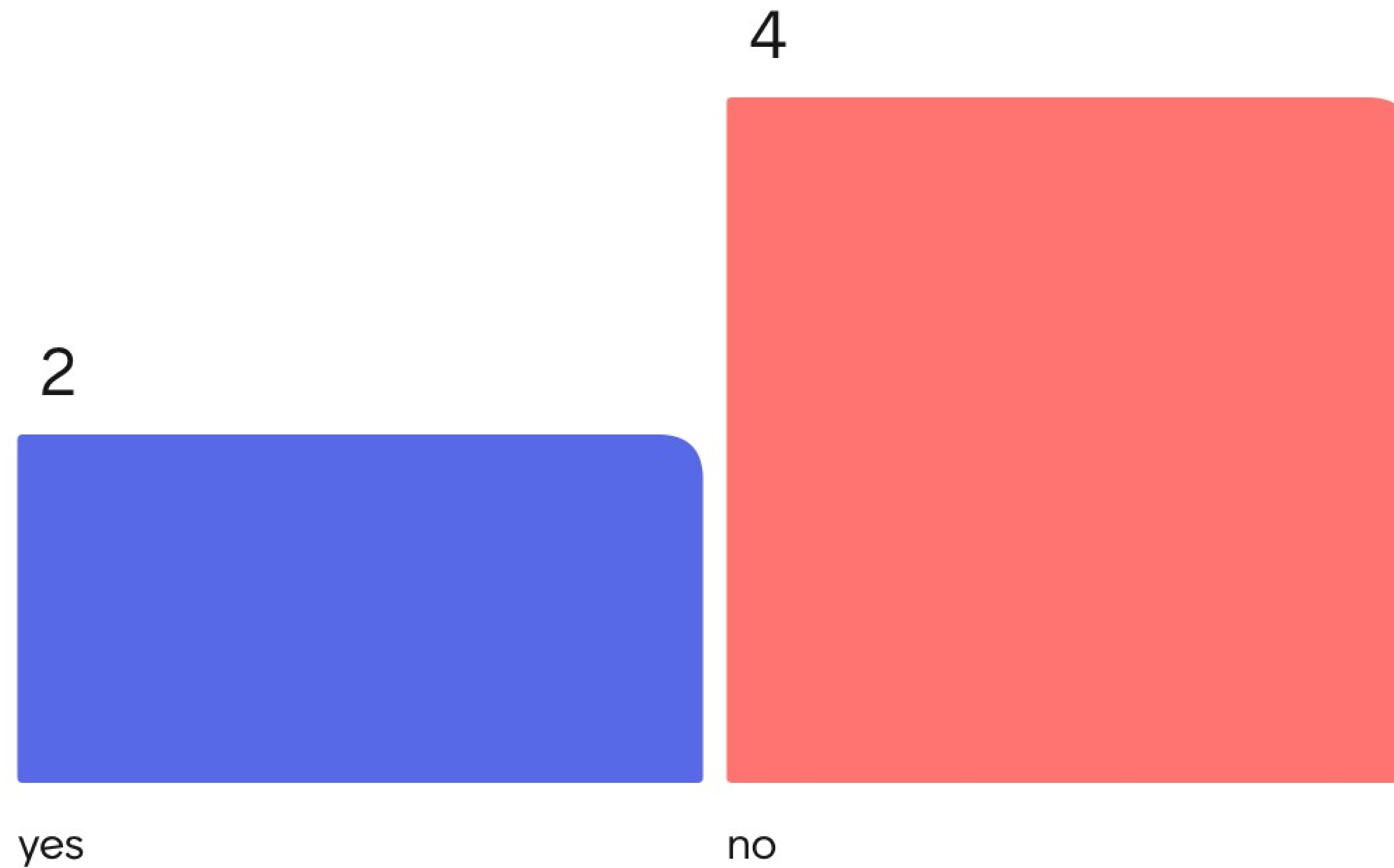
Design principles are high-level, abstract guidelines for creating good software, offering a "why" for good design, such as the SOLID principles which promote maintainable and flexible systems.

Design patterns are concrete, reusable solutions to specific, recurring design problems, providing a "how-to" for implementing those principles, such as the Singleton or Observer patterns.

Principles guide overall architecture, while patterns offer specific implementation details and have defined use cases, working together to create robust software.

10 mins break

Have you heard about **General Responsibility Assignment Software Patterns**?

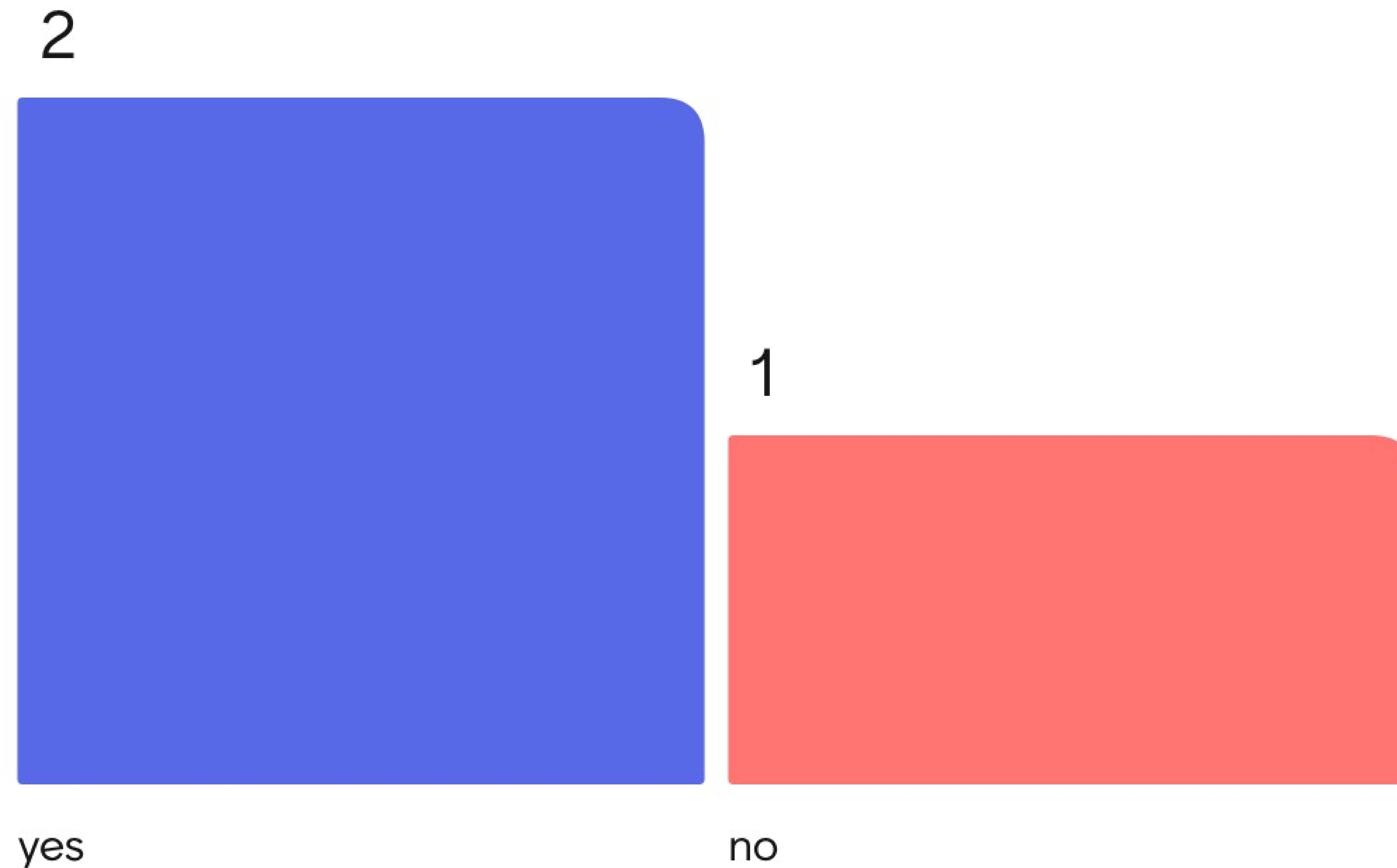


Learn

[[https://en.wikipedia.org/wiki/GRASP_\(object-oriented_design\)](https://en.wikipedia.org/wiki/GRASP_(object-oriented_design))]

- Creator
- Expert
- Controller
- Loose coupling
- High cohesion

Have you looked through the practice part of the topic?



What is your expectation about this course?

Q&A part

0 questions

0 upvotes