

# How's your final project going? Have you started drawing your UML?

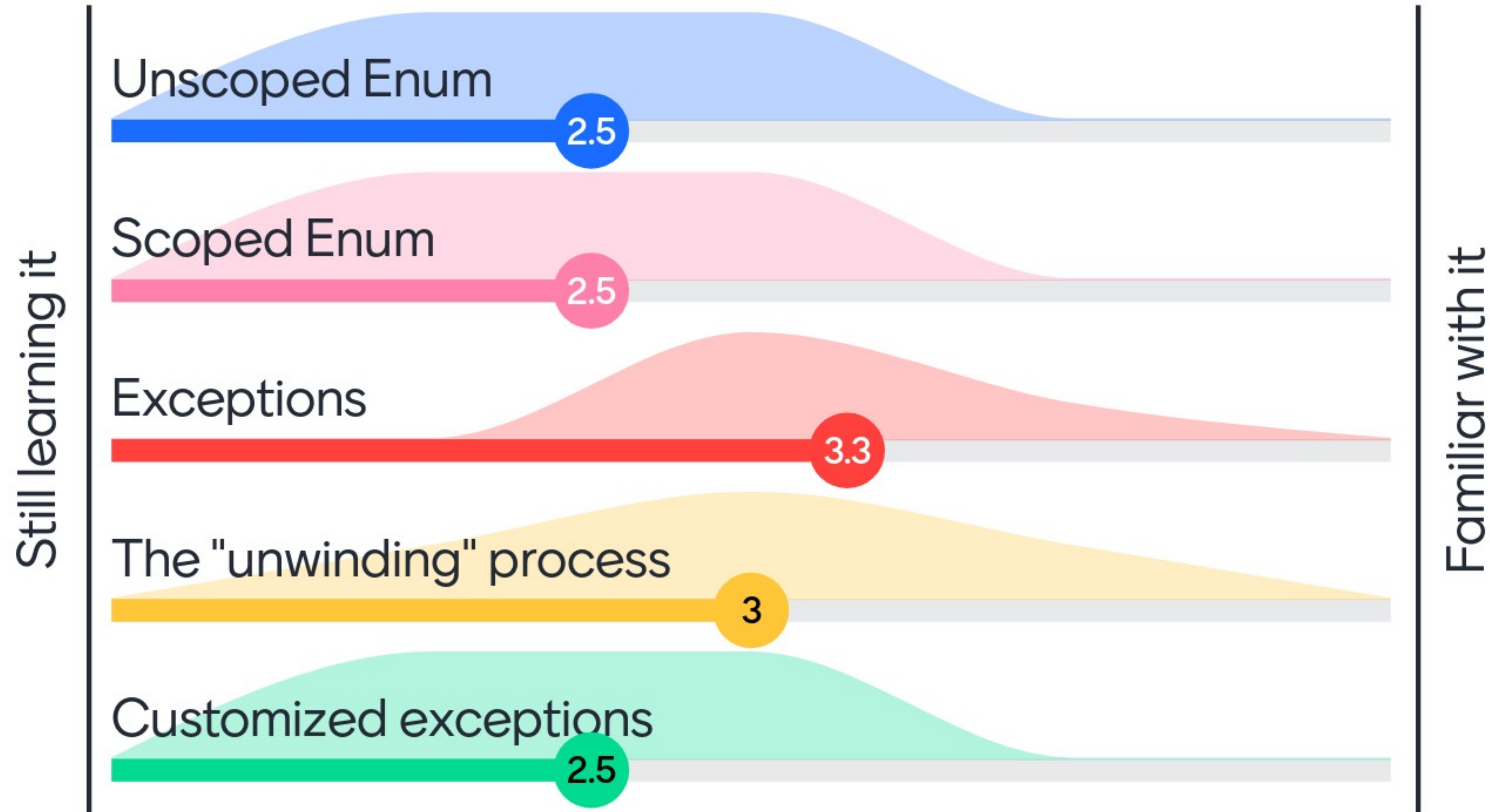
not yet  
starting soon

# Today's plan

- Recap questions
- In-class Ex 13-1



# How much do you know about these topics?





# What is the difference between an unscoped and scoped enum?

For unscoped, values can be converted to their underlying type implicitly but has to be explicitly for scoped



dont worry about conflicts



the ability to compare different enums




unscoped enum can be converted to it's original type implicitly



The correct answer is: Unscoped enum is implicitly converted to an integer for comparison/assignment while scoped one requires explicit conversion (strong type, safer).

# Why do we use exceptions?

To find and handle cases where fatal errors are thrown and to prevent program from crashing 

deal with error cases more flexibly 

The correct answer is: To indicate a fatal error occurred where there is no reasonable way to continue from the point of the error (but might be a way to continue elsewhere)



What keyword is used to 1) generate an exception; 2) indicate the block of code may generate an exception; 3) indicate what to do for the exception?

1) throw, 2) try, 3) catch



Try, Catch



throw try catch



The correct answer is: `throw`, `try`, and `catch`

# In the case of multiple matching **catch** blocks, which one catches the exception?

The first catch where the type of exception either equals or is a base class of the exception that was thrown



the first one



the first one that matches the specific or base type of error



The correct answer is: The first one that matches the exception type (including the base class type)



# How do you get the message associated with an exception?

e.what()



.what()



The correct answer is: Use the member function `what`, e.g. `e.what()`



# Ask me anything

0 questions

0 upvotes