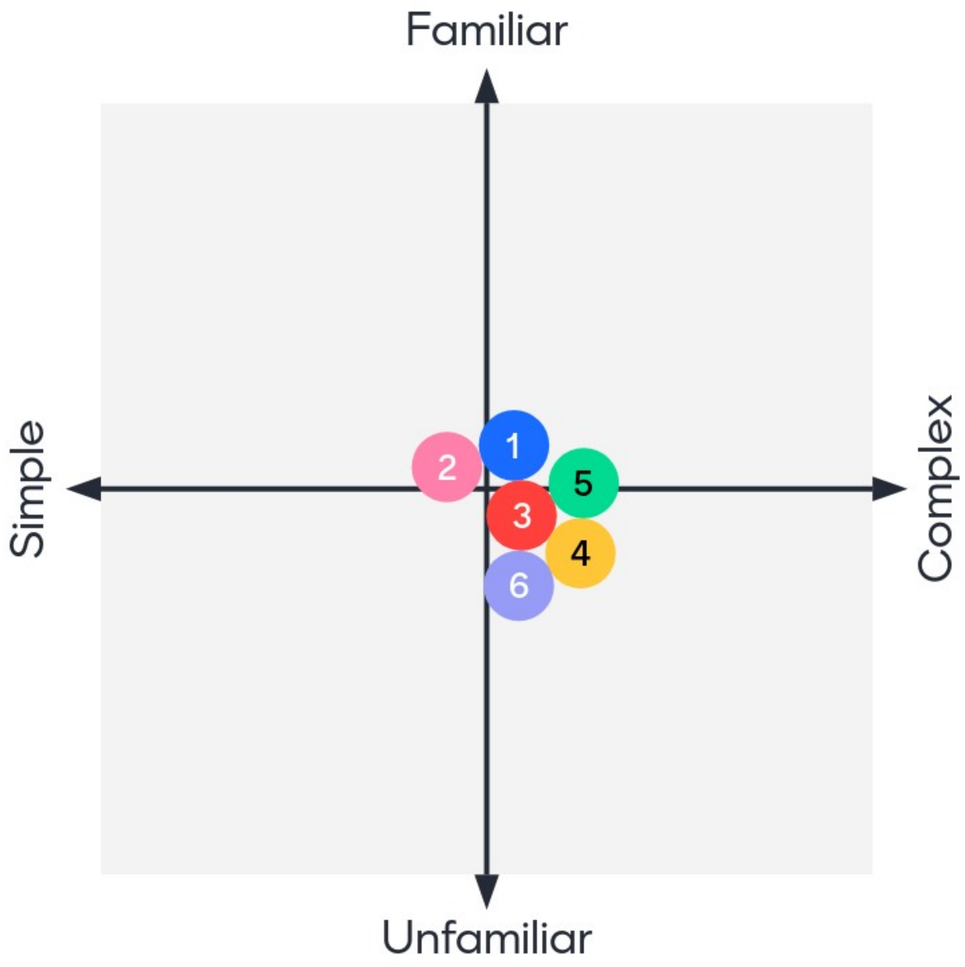
How is your midterm project going so far?

Scale the below topics - Is it simple/complex and how familiar are you with them?

Mentimeter



1 Write a `template struct`

2 Describe what the STL is

3 Replace C arrays with `std::vector`

4 Use `iterator` to iterate an STL container

5 Insert/delete an element to/from `std::vector`

6 Name a few STL containers

Today's plan

- A very brief review and some supplementary
- Recap questions
- In-class exercise 8-1

Standard Template Library (STL)

- **Template**: compiler will replace the placeholder (template parameter) with the actual type when it is used.
- std::vector: a templated data container, a better option than the C array.
- **Iterators**: "a clever pointer" that can iterate through the data container.
- for(std::xxx::iterator it = yyy.begin(); it != yyy.end(); ++it) { ... }
- *it: looks like "dereferencing", but it actually is an operator function overloaded to get the value.
- std::vector::front() and std::vector::back() to get the first and last element in the vector container.
- std::vector::push_back vs std::vector::emplace_back: implicit vs explicit

What can you use template for in C++?

Mentimeter

- Way of writing an object (Node) or function so it can work with variables of any type ✗
- Support for multiple types ✗
- As a placeholder for data type; write function that can be applied to all data types ✗
- array vector set list map stack deque ✗

The correct answer is: To generalize functions or later on classes so they can work with *any* type.

4

What is the STL?

Mentimeter

A set of template classes (contains iterators, containers, algorithms) ✕

A library of additional types for c++ ✕

standard template library . a cpp library of useful data structure and algos ✕

Standard Template Library ✕

The correct answer is: Standard template library. It is a collection of standardly used, templated functions and classes.

4

Given **std::vector<int> vec = {1,2,3};**, write codes to iterate **vec** and print the element's values using **iterator**.

Mentimeter

for(vector<int>::iterator it = vec.begin(); it != vec.end(); ++it) {std::cout>> vec[i];}

for(vector<int>::iterator it = vec.begin(); it != vec.end(); ++it) { cout << *it << endl; }

for(std::vector<int>::iterator t; t != vec.end(); t++)

for (vector<int>::iterator it=vec.begin();it!=vec.end();it++) {cout<<*it<<endl;}

The correct answer is: `for (std::vector<int>::iterator it = vec.begin(); it != vec.end(); ++it) std::cout << *it << std::endl;`

4

Given **std::vector<double> vec;**, how do you add an element at the end to **vec**?

Mentimeter

- vec.push_back(d); ✗
- vec.push_back(element); ✗
- vec.push_back() ✗
- vec.push_back("0.1"); ✗
- vec.pushback(); ✗

The correct answer is: vec.push_back(val); or vec.emplace_back(val); (vec.insert(vec.back(),val);)

5

What is the output of the program below?

Mentimeter

first == 4.5, middle1 == 0.5, middle2 == 4, last == 20

0.5\n 3.5\n 8\n 10\n

first == 4.5/n middle1 == 0.5/n middle2 == 4.0/n last ==20.0

1 5 12 20

The correct answer is: first == 4.5\nmiddle1 == 0.5\nmiddle2 == 4\nlast == 20\n

4

Ask me anything

Mentimeter

0 questions

0 upvotes