Midterm project is done. Five more to go. Which part of C do you like best?

Mentimeter

dynamic memory

anything except recursion

2

Today's plan

- Review ex 9-1
- Recap questions
- In-class ex 9-2

Ex 9-1: Revisit of command line arguments

- int main(int argc, char **argv);
- argc: number of arguments (including the exe)
- argv: an array of input arguments (c strings)
- argv[0]: the exe name
- argv[i]: the ith arguments (separated by whitespaces)

Ex 9-1: Input from command line arguments

- ./abbrev input.txt output.txt
- argc = 3
- argv[0] = "./abbrev"
- argv[1] = "input.txt"
- argv[2] = "output.txt"

Ex 9-1: Read/write files using **std::fstream**

- std::ifstream input(argv[1])
- std::ofstream output(argv[2])
- Check error status using: input.is_open() and output.is_open()
- At the end, it will be nice to close it: input.close() and output.close()
- Yet, the destructor of ifstream and ofstream will also do that for us when the objects are destroyed

Ex 9-1: Reading line by line into **std::string**

- std::string line
- while (getline(input, line)) {...}
- getline will read a line from the stream input

Ex 9-1: Parsing a line using **stringstream**

- std::stringstream ss(line)
- This create a stringstream with line loaded into the buffer
- std::string word
- while (ss >> word) {...} to parse each token in the line
- Now we have word contains one word from the input file

Ex 9-1: Replace vowels with apostrophe (loop)

- Loop the word and check if each character is a vowel
- Replace consecutive vowels with apostrophe
- std::string result; bool isVowel = false;
- for (int i = 0; word[i]; ++i) {
- if(word[i] == 'a' || word[i] == 'A' || ...) isVowel = true;
- else {
- if (isVowel) { result.push_back('\''); isVowel = false;}
- result.push_back(word[i]);
- }
- }
- if (isVowel) result.push_back('\'');

Ex 9-1: Replace vowels with apostrophe (regex)

- #include <regex>
- Don't need to parse the line
- Setup the regex: std::regex reg("[aeiouAEIOU]+");
- Regex replace: std::regex_replace(line, reg, "'");

Ex 9-1: **string** to a particular **data type**

- Reading from std::cin token by token: while(std::cin >> token)
- Create a std::stringstream with token: std::stringstream ss(token)
- Extract from a stringstream and check if it is extracted:
- int i; double d;
- if ((ss >> i) && s.eof()) { // converted the whole string to an int }
- else { // token is not an int}
- Reset the stream position: ss.seekg(0)
- if ((ss >> d) && s.eof()) { // converted the whole string to a double }
- else { // token is not a double}

Ex 9-1 (optional): frequency

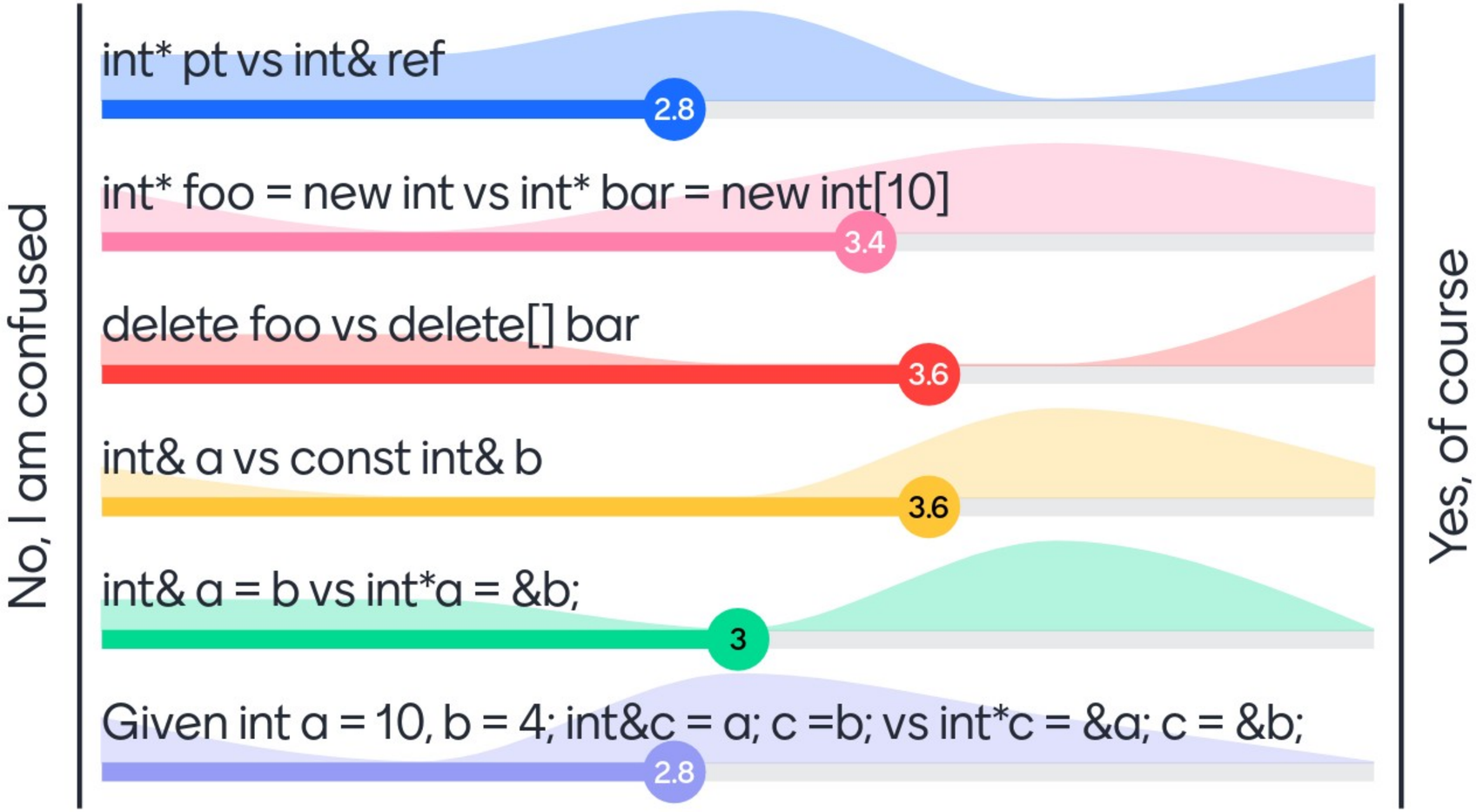
- Use the Bucket structure
- std::vector< Bucket > freq(26)
- Init frequency
- for (int i = 0; i < 26; ++i) { freq[i].letter = 'a' + i; freq[i].count = 0; }
- Use std::ifstream::getc to get a char and isalpha to check if it is an alphabet
- char c; std::ifstream input(filename);`
- `while (input.get(c)) { if (isalpha(c) { // increase the count } }

Ex 9-1 (optional) : customized **std::sort**

- Implement a 'comparator': bool compare_buckets(const Bucket& left, const Bucket& right);
- sort: most frequent to least frequent, if tie, sort by the letter
- if (left.count == right.count) return left.letter < right.letter;
- else return left.count > right.count;

Can you tell the difference between them?

Mentimeter



5

What is a C++ reference?

Mentimeter

an alias variable, which refers to the same memory address of an variable

An alias for a variable

The correct answer is: An alias for another existing variable.

2

What are the three main differences between a pointer and a reference?

Mentimeter

to get the value from a pointer, one has to dereference it

We can't reassign a variable to a reference variable, but we can reassign a memory address to a pointer

A reference directly accesses the variable, doesnt contain the address of it. must be initiailzed first.

reference does not need to be dereferenced, does not store memory address,

The correct answer is: 1. Can't be NULL, 2. must be initialized when defined, 3. can't be changed.

4

When should you use C++ references?

Mentimeter

most of the time

When we want to change a variable passed into a function

Whenever you can to replace pointers. Passing into functions.

instead of pointers, to pass by reference

The correct answer is: Use it when you need to pass a variable by reference and change it inside. However, if need to reallocate, use pointers.

4

How do you allocate memory in C++?

Mentimeter

new
2x

new int; new int[10];

new keyword and deallocate with delete

The correct answer is: new / new []

4

How do you free memory in C++?

Mentimeter

- delete / delete[] ✓
- delete ✗ 3x
- delete var; delete array[] ✗

The correct answer is: delete / delete[]

5

Ask me anything

Mentimeter

0 questions

0 upvotes