ISTANBUL TECHNICAL UNIVERSITY
ITU ZES Solar Car Team

2024/2025

Weekly Assignment-2

24.12.2024

1 C++ Variables

In this chapter you are expected to search variable types in C++ in detail. You need to write a program that creates all possible combinations of int, float, and double with long, short, unsigned keywords. After that, the program must print these properties of variables:

- Print the sizes of all variables using std::sizeof()
- Create pointers to all the variables you've written, then print the sizes of these pointers also
- Print the max and min values of each variable using std::numeric_limits()
- Also explain auto variables, const, and constexpr keywords

You must write a report covering all of your searches about each variable. Also search int(N)_t(for example int32_t), float(N)_t(for example float64_t) variable types, and explain when to use fixed_with int and float types. Do not forget to include size_t variable and wchar_t variable types, and write the differences between these types with respectively int and char. You can add the terminal outputs of your code to the report. You must submit both the report and C++ code of your work.

For the output of your project, you may need to format your output using the "`<iomanip>`" library of C++. You may refer to the below visuals for example output format:

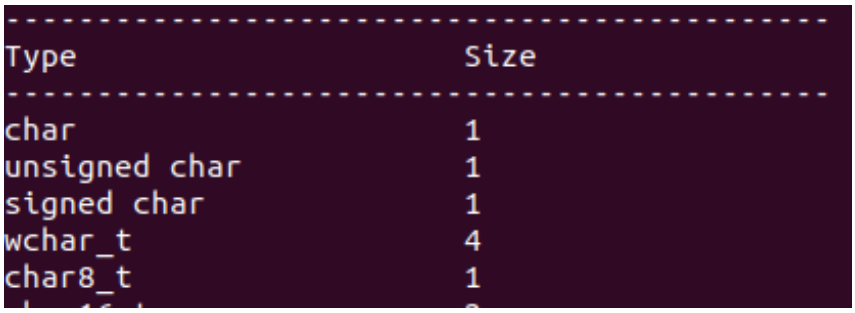

Figure 1: Types and Sizes

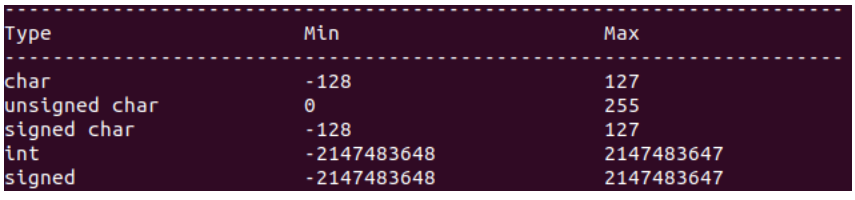

Figure 2: Types and Their Ranges

2 C++ Pointers

In this chapter you are expected to further research the pointers in C++. During this week, you can read documentation, watch videos, or use any material to learn what the pointers are and what their pros are in C++ codes. You need to prepare an oral presentation so that you explain your research and what you learned about pointers during this week.

Also, you need to write a report to describe some topics given below about pointers:

- Garbage Collection Mechanism
- Smart Pointers(void pointer, nullptr, auto_ptr, unique_ptr, shared_ptr, weak_ptr)
- Cons of raw pointers, you need to further search memory leaks, dangling pointers, wild pointers, data inconsistency, and buffer overflow.
- When to use each one of the smart pointers and write the ownership model of each pointer in memory

Moreover, you are expected to write a cpp file containing your applications with pointers and smart pointers. In this source code you are expected to write a program that contains such implementations:

- Create your own SmartPointer class and implement constructor, destructor, operator* overload, operator-> overload of that smart pointer. Test methods you implemented in your main function.
- Add std::auto_ptr variable and use get() function of this pointer to get the memory pointed by the pointer. Also, try move() function of that pointer to move memory the pointer pointed to another pointer instance.
- Add std::unique_ptr variable and use get() and move() function as above. Explain why auto_pointer is deprecated and uniqur_ptr is used in place of it.
- Add std::shared_ptr variable and use get() and move() functions, also use use_count() function of shared pointer. Explain the sharing system by implementing more than one instance and reset() function.
- Add std::weak_ptr variable and use get() and move() function. Explain what cyclic dependency is and why weak_ptr s are used.

For coding implementations of these std magic pointers you must print the memory addresses that the pointer points to after each function you used for changing or resetting the memory address of the pointer.

You are expected to write a report including your search, and also submit source files you created. You can add output screenshots of your code to your report.

3 Conclusion

For any questions and errors you find, contact Nisanur Çontay. While preparing your work consider readability and scalability, and also do not forget to add necessary comments to your codes.