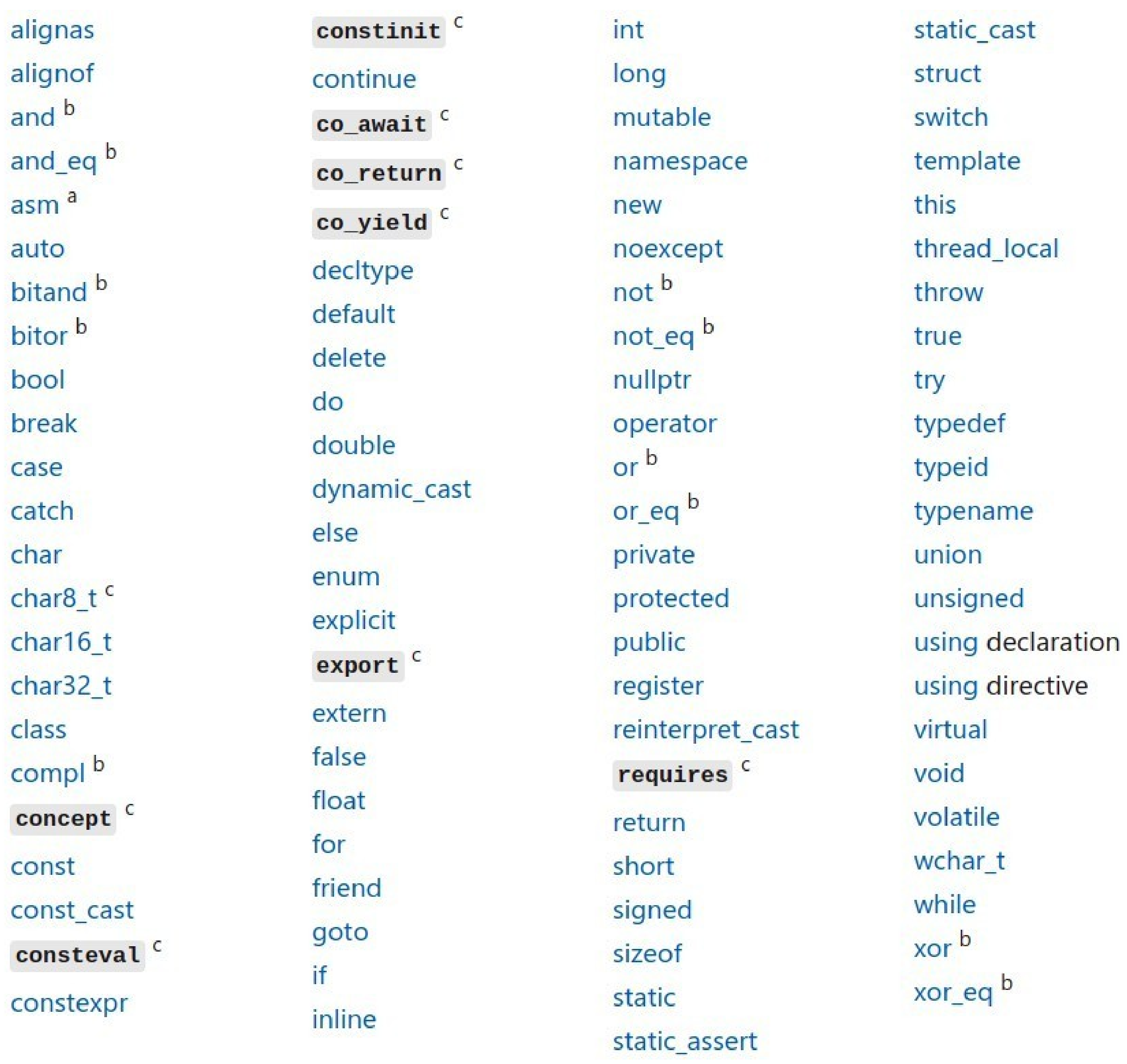
In the previous Lab, we have used multiple FSA to identify different type of tokens. In this lab you are required to implement Look A Head logic in Lexical Analyzer.

There will be FSA for Numbers, Floating Point Number, String and identifier. Your program will read source file character by character and identify different type of tokens. Following special characters will serve as separator (a special sequence of character which will denote End of a token and start of a new token).

Space, “\t”, “\n”, +, - , *, / , %, ++, - -, =, +=, -=, *=, /=, %=, ==, !=, >, < , >=, <=, &&, ||, !, &, |, ^, ~, <<, >>, ?:, ., - >,

For your reference following is the list of keyword.



alignas	constinit [c]	int	static_cast
alignof	continue	long	struct
and [b]	co_await [c]	mutable	switch
and_eq [b]	co_return [c]	namespace	template
asm [a]	co_yield [c]	new	this
auto	decltype	noexcept	thread_local
bitand [b]	default	not [b]	throw
bitor [b]	delete	not_eq [b]	true
bool	do	nullptr	try
break	double	operator	typedef
case	dynamic_cast	or [b]	typeid
catch	else	or_eq [b]	typename
char	enum	private	union
char8_t [c]	explicit	protected	unsigned
char16_t	export [c]	public	using declaration
char32_t	extern	register	using directive
class	false	reinterpret_cast	virtual
compl [b]	float	requires [c]	void
concept [c]	for	return	volatile
const	friend	short	wchar_t
const_cast	goto	signed	while
consteval [c]	if	sizeof	xor [b]
constexpr	inline	static	xor_eq [b]
		static_assert	

Identifier is start with _, a-z or A-Z and their can be repetition of number and a-z A-Z _.

Numbers and Floating point number.
String is start with “ and ends with “.
Operator can be seen on https://docs.microsoft.com/en-us/cpp/cpp/cpp-built-in-operators-precedence-and-associativity?view=msvc-170

Punctuation are https://docs.microsoft.com/en-us/cpp/cpp/punctuators-cpp?view=msvc-170