**1. Основы C++ (1-2 недели)**

**Теория:**

* Ввод-вывод (cin, cout) - OK
* Типы данных (int, float, double, char, bool) - OK
* Операторы (+, -, \*, /, %, ++, --, ==, !=, >, <, &&, ||, !) - OK
* Условные операторы (if, else, else if, switch-case) - OK
* Циклы (for, while, do-while) – OK.

**Практика:**

1. Калькулятор (сложение, вычитание, умножение, деление). - OK
2. Проверка числа на чётность. - OK
3. Вывод таблицы умножения (например, для числа 5). - OK
4. Поиск наибольшего из трёх чисел. - OK
5. Проверка, является ли число простым. - OK

**2. Массивы и строки (1-2 недели)**

**Теория:**

* Одномерные и многомерные массивы -ok
* Строки (std::string, C-строки) -ok
* Алгоритмы поиска и сортировки (линейный поиск, пузырьковая сортировка) - ok

**Практика:**

1. Найти сумму элементов массива. - OK
2. Поиск максимального/минимального элемента в массиве.
3. Перевернуть массив (реверс).
4. Подсчёт количества гласных букв в строке.
5. Проверка, является ли строка палиндромом.

**3. Функции и рекурсия (1-2 недели)**

**Теория:**

* Объявление и вызов функций
* Передача параметров (по значению, по ссылке, по указателю)
* Рекурсия

**Практика:**

1. Функция, вычисляющая факториал числа (итеративно и рекурсивно).
2. Функция, проверяющая, является ли число степенью двойки.
3. Реализация алгоритма Евклида (НОД).
4. Числа Фибоначчи (рекурсия + мемоизация).
5. Быстрое возведение в степень (рекурсивно).

**4. Указатели и динамическая память (1-2 недели)**

**Теория:**

* Указатели и ссылки
* Динамическое выделение памяти (new, delete)
* Утечки памяти

**Практика:**

1. Обмен значений двух переменных через указатели.
2. Создание динамического массива и его сортировка.
3. Реализация функции strcpy (копирование строки).
4. Подсчёт длины строки через указатели.
5. Двумерный динамический массив (матрица).

**5. ООП (2-4 недели)**

**Теория:**

* Классы и объекты
* Инкапсуляция, наследование, полиморфизм
* Конструкторы и деструкторы
* Перегрузка операторов

**Практика:**

1. Класс "Точка" с методами для вычисления расстояния до другой точки.
2. Класс "Стек" (реализация LIFO).
3. Класс "Дробь" с перегрузкой операторов +, -, \*, /.
4. Иерархия классов "Фигура" → "Прямоугольник", "Круг".
5. Виртуальные методы и полиморфизм (пример с разными типами животных).

**6. STL и работа с файлами (1-2 недели)**

**Теория:**

* Контейнеры (vector, list, map, set)
* Итераторы
* Работа с файлами (ifstream, ofstream)

**Практика:**

1. Подсчёт частоты слов в тексте (используя map).
2. Сортировка вектора структур (например, по возрасту).
3. Чтение чисел из файла и запись их квадратов в другой файл.
4. Поиск дубликатов в массиве с помощью set.
5. Реализация простого словаря (слово → перевод).

**7. Продвинутые темы (многопоточность, шаблоны, исключения) (2-4 недели)**

**Теория:**

* Многопоточность (std::thread)
* Шаблоны функций и классов
* Обработка исключений (try, catch, throw)

**Практика:**

1. Параллельное вычисление суммы элементов массива.
2. Шаблонная функция для поиска максимума в массиве любого типа.
3. Обработка деления на ноль через исключения.
4. Реализация простого thread-safe счетчика.
5. Шаблонный класс "Очередь" (FIFO).

**8. Проекты для закрепления (1-2 месяца)**

1. Консольная игра "Крестики-нолики".
2. Мини-база данных студентов (CRUD).
3. Чат-сервер и клиент (сокеты, многопоточность).
4. Упрощённый аналог grep (поиск текста в файлах).
5. Свой собственный vector (аналог std::vector).

**Дополнительные ресурсы:**

* **Книги:**
  + Бьерн Страуструп - "Язык программирования C++"
  + Стивен Прата - "C++ Primer"
* **Платформы для практики:**
  + LeetCode, Codeforces, HackerRank
  + Codewars (ката для начинающих)