**🧩 Объектно-ориентированное программирование (ООП)**

* Что такое ООП?
* В чем разница между функциональным и объектно-ориентированным программированием?
* Основные принципы ООП:
  + Инкапсуляция
  + Наследование
  + Полиморфизм

**🏗️ Классы**

* Что такое класс?
* Чем класс отличается от структуры?
* Что такое объект класса?
* Сколько «весит» класс?
* Что такое спецификаторы доступа и какие они бывают?
* Что такое функция-аксессор (getter/setter)?
* Для чего используется ключевое слово this?

**⚙️ Конструкторы и деструкторы**

* Что такое конструктор?
* Виды конструкторов.
* Что такое делегирование конструкторов?
* Что такое список инициализации?
* Для чего нужен деструктор и какие бывают его виды?

**🧭 Статические и константные члены класса**

* Что такое статическое поле класса?
* Особенности создания и использования статических членов класса.
* Что такое константное поле и как его можно проинициализировать?
* Можно ли сделать метод константным?

**➕ Перегрузка операторов**

* Что такое перегрузка операторов?
* Что такое дружественная функция?
* Какие существуют способы перегрузки операторов?
* Почему важно правильно перегружать оператор присваивания?
* Всегда ли необходимо перегружать операторы?
* Как перегрузить оператор << для cout?

**🧩 Шаблоны (Templates)**

* Для чего нужны шаблонные классы и функции?
* Принцип работы шаблонных функций.
* Почему могут возникать ошибки линковки при работе с шаблонными классами?

**📊 Структуры данных (Data Structures)**

* Что такое структура данных?
* В чем разница между линейными и нелинейными структурами данных?
* Что такое массив (статический и динамический)?
* Что такое список и какие бывают его виды?
* Что такое очередь и какие бывают ее виды?
* Что такое стек и дек?
* Что такое дерево и какие бывают виды деревьев?
* Что такое балансировка дерева и почему она важна?
* Особенности бинарного дерева поиска (BST).
* Что такое хеш-таблица?

**🧬 Наследование**

* Что такое вложенные классы и зачем они нужны?
* Что такое отношения между классами: агрегация и композиция?
* Какие типы наследования существуют в C++?
* Особенности работы конструкторов и деструкторов при наследовании.
* Что такое множественное наследование и почему оно может быть проблемным?

**🌀 Полиморфизм**

* Типы полиморфизма: статический и динамический.
* Что такое раннее и позднее связывание?
* Что такое указатель или ссылка на базовый класс?
* Что такое виртуальный метод?
* Ключевые слова virtual и override.
* В чем разница между перегрузкой (overload) и переопределением (override) метода?
* Что такое таблица виртуальных методов (vtable) и указатель vptr?
* Что такое абстрактный класс и как его создать?
* Что такое чисто виртуальный метод?
* Для чего нужен виртуальный деструктор?
* Для чего нужен чисто виртуальный деструктор?

**🚨 Исключения (Exceptions)**

* Что такое исключение?
* Зачем нужны ключевые слова try, catch и throw?
* Что такое базовый класс exception?
* Для чего нужна иерархия исключений?

**🧠 RTTI (Run-Time Type Information)**

* Для чего используется оператор typeid?
* Какие виды преобразований существуют в C++?
* Как работают:
  + dynamic\_cast
  + static\_cast

**💾 Потоки (Streams)**

* Что такое поток ввода и его виды?
* Что такое поток вывода и его виды?
* Как работают файловые потоки?

**📚 STL (Standard Template Library)**

* Что такое контейнер? Какие бывают виды контейнеров?
* Что такое итератор? Какие виды итераторов вы знаете?
* Что такое алгоритм?
* Что такое предикат?
* Что такое lambda-выражение?
* Что такое умный указатель и какие бывают его виды?

**🧱 Классы STL**

* string
* array
* vector
* list
* deque
* stack
* queue
* priority\_queue
* set, multiset
* map, multimap
* unordered\_set, unordered\_multiset
* unordered\_map, unordered\_multimap