

Курс "Программирование на C/C++: от основ к ООП"

1-й модуль

Цель курса: дать подросткам, знакомым с Python, глубокие знания языков C и C++, научить их писать эффективный и оптимизированный код, решать сложные алгоритмические задачи и заложить прочный фундамент для дальнейшего развития в IT-сфере, включая подготовку к олимпиадам и поступлению в технические вузы.

Программа курса:

День первый

Введение в язык C. Первая программа

- Знакомство с историей и стандартами языка C;
- Обсуждение этапов компиляции: от исходного кода к исполняемому файлу;
- Установка и настройка рабочего окружения (компилятор GCC, редактор VS Code);
- Написание, компиляция и запуск первой программы "Hello, World!".

Результат занятия: ученики познакомятся с языком C, настроят рабочее окружение, поймут процесс компиляции и напишут свою первую программу.

Практическое задание: самостоятельно написать программу, которая выводит на экран информацию о себе (имя, возраст, хобби).

День второй

Переменные, типы данных и системы счисления

- Изучение основных систем счисления (двоичная, шестнадцатеричная);
- Переменные: объявление, инициализация;
- Базовые типы данных: int, char, float, double;
- Модификаторы типов: short, long, unsigned, signed;
- Операция sizeof для определения размера типа в памяти.

Результат занятия: ученики научатся работать с переменными и различными типами данных, поймут, как данные хранятся в памяти компьютера.

Практическое задание: создание программы-калькулятора для простых арифметических операций с двумя целыми числами и вводом чисел с клавиатуры.

День третий

Базовые операции ввода-вывода

- Стандартные потоки ввода, вывода и ошибок;
- Посимвольный ввод-вывод: getchar() и putchar();
- Форматированный вывод с помощью printf;
- Форматированный ввод с помощью scanf(): получение данных от пользователя.

Результат занятия: освоили основные функции для взаимодействия с пользователем через консоль, научатся выводить данные в нужном формате и считывать пользовательский ввод.

Практическое задание: создать программу-анкету, которая запрашивает данные пользователя и выводит информацию в виде красивого отформатированного текста.

День четвертый

День четвертый: Арифметические и логические операции

- Арифметические операторы: +, -, *, /, % (остаток от деления);
- Операторы инкремента (++) и декремента (--);
- Операторы сравнения (==, !=, >, <, >=, <=);
- Логические операторы && (И), || (ИЛИ), ! (НЕ);
- Условный оператор if-else для создания ветвлений в программе.

Результат занятия: ученики научатся использовать арифметические и логические операции для выполнения вычислений и принятия решений в программе.

Практическое задание: создание программы проверки четности числа, с выводом соответствующего сообщения.

Курс "Программирование на C/C++: от основ к ООП"

2-й модуль

Цель курса: дать подросткам, знакомым с Python, глубокие знания языков C и C++, научить их писать эффективный и оптимизированный код, решать сложные алгоритмические задачи и заложить прочный фундамент для дальнейшего развития в IT-сфере, включая подготовку к олимпиадам и поступлению в технические вузы.

Программа курса:

День первый

Сложные условия и побитовые операции

- Оператор множественного выбора `switch`;
- Тернарный условный оператор `?` : как краткая замена `if-else`;
- Введение в побитовые операции: И, ИЛИ, НЕ, XOR (`&`, `|`, `~`, `^`);
- Операторы побитового сдвига `<<` и `>>`.

Результат занятия: ученики научатся использовать `switch` для обработки множества вариантов и освоят побитовые операции для низкоуровневых манипуляций с данными.

Практическое задание: написать программу, которая по номеру месяца (1-12) выводит его название и время года с помощью оператора `switch`.

День второй

Циклы. Конструкции `while` и `for`

- Введение в циклы. Цикл с предусловием `while`;
- Самый популярный цикл `for`: инициализация, условие, итерация;
- Разбор типичных ошибок при работе с циклами (бесконечный цикл).

Результат занятия: ученики поймут, что такое циклы, и научатся использовать `while` и `for` для выполнения повторяющихся действий.

Практическое задание: создать программу, которая выводит таблицу умножения для введенного пользователем числа с помощью цикла `for`.

День третий

Другие виды циклов и управление ими

- Цикл с постусловием `do-while`;
- Вложенные циклы для работы с двумерными структурами;
- Операторы `break` (досрочный выход из цикла) и `continue` (переход к следующей итерации);
- Обсуждение и осторожное использование оператора `goto`.

Результат занятия: ученики изучат все виды циклов в C и научатся управлять их выполнением.

Практическое задание: написать программу, которая с помощью вложенных циклов рисует в консоли прямоугольник из звездочек (*) заданных пользователем ширины и высоты.

День четвертый

Введение в препроцессор

- Задачи препроцессора C;
- Директива `#include` для подключения заголовочных файлов;
- Директива `#define` для создания именованных констант;
- Создание простых макросов с параметрами;
- Условная компиляция (`#ifdef`, `#ifndef`, `#endif`).

Результат занятия: ученики получают представление о препроцессоре и научатся использовать его директивы для улучшения читаемости и гибкости кода.

Практическое задание: создать заголовочный файл с константами (например, `PI`, `E`) и макросом (например, для нахождения максимума из двух чисел), подключить его к основной программе и использовать.

Курс "Программирование на C/C++: от основ к ООП"

3-й модуль

Цель курса: дать подросткам, знакомым с Python, глубокие знания языков C и C++, научить их писать эффективный и оптимизированный код, решать сложные алгоритмические задачи и заложить прочный фундамент для дальнейшего развития в IT-сфере, включая подготовку к олимпиадам и поступлению в технические вузы.

Программа курса:

День первый

Введение в указатели

- Что такое адрес памяти и почему это важно;
- Указатель как переменная для хранения адреса;
- Операция взятия адреса (&) и операция разыменования (*);
- Объявление, инициализация и использование указателей;
- Нулевой указатель NULL.

Результат занятия: ученики поймут одну из ключевых и самых мощных концепций языка C — указатели, и научатся базовым операциям с ними.

Практическое задание: написать программу, которая объявляет переменную, создает указатель на нее, а затем выводит на экран значение переменной, ее адрес в памяти и значение, полученное через разыменование указателя.

День второй

Адресная арифметика и указатели

- Адресная арифметика: как изменяется адрес при сложении и вычитании;
- Связь шага адресной арифметики с размером типа данных;
- Сравнение указателей;
- Приведение типов указателей: зачем и когда это нужно.

Результат занятия: ученики научатся выполнять арифметические операции с указателями для перемещения по памяти, что является основой для работы с массивами.

Практическое задание: создать `int` переменную, указатель на нее, а затем в цикле вывести адреса и значения нескольких следующих и предыдущих ячеек памяти.

День третий

Массивы

- Массив как набор однотипных элементов в памяти;
- Объявление и инициализация одномерных массивов;
- Доступ к элементам массива по индексу;
- Вычисление размера массива;
- Обсуждение отсутствия контроля выхода за границы массива.

Результат занятия: ученики научатся работать с массивами для хранения и обработки коллекций данных.

Практическое задание: написать программу, которая заполняет массив из 10 целых чисел случайными значениями, а затем находит и выводит минимальный и максимальный элементы массива.

День четвертый

Связь массивов и указателей. Строки

- Имя массива как указатель на его первый элемент;
- Передача массивов в функции;
- Использование `const` для защиты данных при передаче в функции;
- Строки как массивы символов, завершающиеся нуль-терминатором (`\0`);
- Объявление и простая инициализация строк.

Результат занятия: ученики поймут фундаментальную связь между массивами и указателями в C и получат базовые навыки работы со строками.

Практическое задание: написать собственную функцию для вычисления длины строки (аналог `strlen`), которая принимает указатель на `char` и подсчитывает символы до нуль-терминатора.

Курс "Программирование на C/C++: от основ к ООП"

4-й модуль

Цель курса: дать подросткам, знакомым с Python, глубокие знания языков C и C++, научить их писать эффективный и оптимизированный код, решать сложные алгоритмические задачи и заложить прочный фундамент для дальнейшего развития в IT-сфере, включая подготовку к олимпиадам и поступлению в технические вузы.

Программа курса:

День первый

Функции для работы со строками

- Стандартная библиотека `<string.h>`;
- Функция `strlen` для определения длины строки;
- Функции `strcpy` и `strncpy` для безопасного копирования строк;
- Функции `strcat` и `strncat` для объединения (конкатенации) строк.

Результат занятия: ученики научатся использовать стандартные функции для выполнения самых частых операций со строками, понимая разницу между безопасными и небезопасными функциями.

Практическое задание: написать программу, которая запрашивает у пользователя имя и фамилию, а затем объединяет их в одну строку "Фамилия, Имя" и выводит на экран.

День второй

Сравнение строк и преобразования

- Функции `strcmp` и `strncmp` для сравнения строк;
- Функция `strstr` для поиска подстроки в строке;
- Функции `atoi`, `atol`, `atof` для преобразования строки в число;
- Функция `sprintf` для форматирования данных в строку.

Результат занятия: ученики расширят свой инструментарий для работы со строками, научатся сравнивать их и преобразовывать в другие типы данных.

Практическое задание: создать простую программу-викторину. Программа задает вопрос, пользователь вводит ответ строкой, а программа сравнивает его с правильным ответом и сообщает результат.

День третий

Основы работы с функциями

- Функции как способ организации и переиспользования кода;
- Объявление (прототип), определение и вызов функции;
- Оператор `return` для возврата значения из функции;
- Тип `void` для функций, которые не возвращают значение.

Результат занятия: ученики поймут назначение функций и научатся декомпозировать программу на более мелкие, управляемые части.

Практическое задание: написать функцию, которая принимает два целых числа в качестве аргументов и возвращает их сумму. Вызвать эту функцию из `main` и вывести результат.

День четвертый

Передача аргументов в функции

- Передача аргументов по значению: что происходит в памяти;
- Передача массивов в функции (по указателю);
- Использование `const` для защиты данных, передаваемых в функцию;
- Практические примеры: функции для поиска среднего значения в массиве, для печати массива.

Результат занятия: ученики поймут разницу в механизмах передачи данных в функции и научатся эффективно работать с массивами внутри функций.

Практическое задание: создать функцию, которая принимает массив целых чисел и его размер, а затем находит и возвращает максимальный элемент этого массива.

Курс "Программирование на C/C++: от основ к ООП"

5-й модуль

Цель курса: дать подросткам, знакомым с Python, глубокие знания языков C и C++, научить их писать эффективный и оптимизированный код, решать сложные алгоритмические задачи и заложить прочный фундамент для дальнейшего развития в IT-сфере, включая подготовку к олимпиадам и поступлению в технические вузы.

Программа курса:

День первый

Указатели на функции и Callbacks

- Синтаксис объявления указателя на функцию;
- Присваивание адреса функции указателю;
- Вызов функции через указатель;
- Концепция "функций обратного вызова" (callback).

Результат занятия: ученики освоят продвинутую, но мощную концепцию указателей на функции, которая позволяет писать более гибкий и универсальный код.

Практическое задание: создать универсальную функцию для обработки массива (например, печати или изменения каждого элемента), которая принимает указатель на функцию, выполняющую действие над элементом.

День второй

Рекурсивные функции

- Понятие рекурсии: функция, вызывающая саму себя;
- Базовый случай (условие выхода из рекурсии) и рекурсивный шаг;
- Риски переполнения стека и как их избежать;
- Классические примеры: вычисление факториала, числа Фибоначчи.

Результат занятия: ученики поймут принцип работы рекурсии и научатся применять ее для решения задач, которые имеют рекурсивную природу.

Практическое задание: написать рекурсивную функцию для вычисления степени числа (a^b).

День третий

Динамическое выделение памяти

- Недостатки статического выделения памяти;
- Стек и куча (heap): где и как выделяется память;
- Функции malloc, calloc, realloc для выделения памяти в куче;
- Функция free для освобождения памяти;
- Проверка на ошибки выделения памяти. Понятие утечек памяти.

Результат занятия: ученики научатся управлять памятью вручную, создавая массивы и структуры данных, размер которых определяется во время выполнения программы.

Практическое задание: написать программу, которая запрашивает у пользователя размер массива, динамически выделяет под него память, заполняет его и выводит на экран, а затем освобождает память.

День четвертый

Структуры (struct)

- Зачем нужны структуры: объединение разнотипных данных в одну сущность;
- Объявление структуры и создание переменных-структур;
- Доступ к полям структуры через оператор ;
- Указатели на структуры и оператор ->;
- Передача структур в функции (по значению и по указателю).

Результат занятия: ученики научатся создавать собственные сложные типы данных для моделирования реальных объектов, что является шагом к объектно-ориентированному мышлению.

Практическое задание: создать структуру `Student` с полями `name`, `age`, `average_grade`. Написать функцию, которая принимает указатель на студента и выводит информацию о нем в консоль.

Курс "Программирование на C/C++: от основ к ООП"

6-й модуль

Цель курса: дать подросткам, знакомым с Python, глубокие знания языков C и C++, научить их писать эффективный и оптимизированный код, решать сложные алгоритмические задачи и заложить прочный фундамент для дальнейшего развития в IT-сфере, включая подготовку к олимпиадам и поступлению в технические вузы.

Программа курса:

День первый

Работа с файлами. Текстовый режим

- Понятие файла и файлового потока. Указатель FILE;
- Открытие и закрытие файла: `fopen` и `fclose`;
- Режимы открытия файла ("`r`", "`w`", "`a`");
- Посимвольное (`fgetc`, `fputc`) и построчное (`fgets`, `fputs`) чтение/запись;
- Форматированный ввод/вывод в файл (`fscanf`, `fprintf`).

Результат занятия: ученики научатся сохранять результаты работы программы в файлы и считывать данные из них, что делает программы гораздо более полезными.

Практическое задание: написать программу, которая записывает в текстовый файл несколько строк, введенных пользователем, а затем закрывает файл. Написать вторую программу, которая читает этот файл и выводит его содержимое на экран.

День второй

Работа с файлами. Бинарный режим

- Отличие бинарного режима от текстового.
- Функции `fread` и `fwrite` для чтения и записи блоков данных.
- Запись и чтение структур в бинарный файл.
- Функции для управления позицией в файле: `fseek`, `ftell`.

Результат занятия: ученики освоят работу с бинарными файлами, что позволяет эффективно хранить сложные структуры данных.

Практическое задание: модифицировать программу со структурой `Student`. Создать массив студентов, записать его в бинарный файл, а затем прочитать из файла и вывести на экран.

День третий

Введение в C++. Первые отличия

- Философия C++: "лучший C".
- Первая программа на C++. Поточный ввод/вывод с `cin` и `cout`.
- Пространства имен (`namespace`) для избежания конфликтов имен.
- Директива `using`.
- Новый тип данных `bool`.

Результат занятия: ученики сделают первый шаг в мир C++, увидят его основные отличия и преимущества в плане удобства написания кода.

Практическое задание: переписать программу-анкету из 1-го модуля с использованием `cin` и `cout` вместо `scanf` и `printf`.

День четвертый

Ссылки и класс `std::string`

- Ссылки (&) как псевдонимы для переменных. Их отличия от указателей.
- Передача аргументов в функции по ссылке.
- Введение в класс `std::string` из стандартной библиотеки.
- Создание строк, конкатенация, определение длины и доступ к символам.

Результат занятия: ученики познакомятся с мощными инструментами C++ — ссылками и классом `string`, которые значительно упрощают работу по сравнению с C.

Практическое задание: переписать программу для объединения имени и фамилии из 4-го модуля с использованием `std::string`.

Курс "Программирование на C/C++: от основ к ООП"

7-й модуль

Цель курса: дать подросткам, знакомым с Python, глубокие знания языков C и C++, научить их писать эффективный и оптимизированный код, решать сложные алгоритмические задачи и заложить прочный фундамент для дальнейшего развития в IT-сфере, включая подготовку к олимпиадам и поступлению в технические вузы.

Программа курса:

День первый

Функции в C++ и файловые потоки

- Перегрузка функций: создание функций с одинаковыми именами, но разными параметрами;
- Аргументы функций по умолчанию;
- Файловые потоки C++: ifstream (чтение), ofstream (запись), fstream;
- Открытие, чтение, запись и закрытие файлов в стиле C++.

Результат занятия: ученики изучат продвинутые возможности функций в C++ и научатся работать с файлами с помощью удобных файловых потоков.

Практическое задание: написать программу, которая копирует содержимое одного текстового файла в другой, используя файловые потоки C++.

День второй

Основы ООП. Классы и объекты

- Отличие структур в C++ от классов. Ключевое слово class;
- Инкапсуляция: объединение данных и методов;
- Модификаторы доступа: public и private;
- Создание объектов (экземпляров класса);
- Методы класса (функции-члены).

Результат занятия: ученики познакомятся с фундаментальной концепцией ООП — классами, и научатся создавать собственные типы данных с закрытыми данными и публичными методами.

Практическое задание: создать класс `Rectangle` (прямоугольник) с приватными полями `width` и `height` и публичными методами для установки размеров и вычисления площади.

День третий

Конструкторы и деструкторы

- Конструктор для инициализации объектов класса;
- Конструктор по умолчанию и конструкторы с параметрами;
- Деструктор для освобождения ресурсов, занимаемых объектом;
- Порядок вызова конструкторов и деструкторов.

Результат занятия: ученики научатся использовать конструкторы для правильной инициализации объектов и деструкторы для корректного завершения их работы.

Практическое задание: добавить в класс `Rectangle` конструктор, который будет инициализировать ширину и высоту при создании объекта. Добавить в конструктор и деструктор вывод сообщений в консоль, чтобы отследить их вызов.

День четвертый

Динамическая память в C++ и умные указатели

- Операторы `new` и `delete` для выделения и освобождения памяти в куче;
- Проблема ручного управления памятью: утечки и двойное освобождение;
- Введение в умные указатели: `std::unique_ptr`;
- `std::unique_ptr` как безопасный способ владения ресурсом;
- Обсуждение и выбор темы для итогового проекта.

Результат занятия: ученики познакомятся с современным и безопасным подходом к управлению памятью в C++ с помощью умных указателей и спланируют свой первый небольшой проект.

Практическое задание: переписать программу с динамическим выделением памяти из 5-го модуля с использованием `new/delete`, а затем с `std::unique_ptr`, чтобы увидеть разницу.

Курс "Программирование на C/C++: от основ к ООП"

8-й модуль

Цель курса: дать подросткам, знакомым с Python, глубокие знания языков C и C++, научить их писать эффективный и оптимизированный код, решать сложные алгоритмические задачи и заложить прочный фундамент для дальнейшего развития в IT-сфере, включая подготовку к олимпиадам и поступлению в технические вузы.

Программа курса:

День первый

Проектирование и архитектура

- Декомпозиция задачи: разбиение большого проекта на мелкие части;
- Создание заголовочных файлов (.h) для объявления функций и структур;
- Разделение кода на несколько .c или .cpp файлов;
- Обсуждение архитектуры выбранного проекта.

Результат занятия: ученики научатся правильно структурировать проект, разделяя интерфейс и реализацию, и спроектируют архитектуру своего итогового проекта.

Практическое задание: создать структуру файлов и заголовочные файлы для своего проекта. Написать `main` файл, который пока вызывает пустые функции.

День второй

Реализация основного функционала

- Написание кода для ключевых функций проекта;
- Пошаговая отладка с помощью отладчика в VS Code;
- Работа с точками останова (breakpoints) и просмотр значений переменных.

Результат занятия: ученики реализуют ядро своего проекта и освоят базовые навыки отладки кода, что является критически важным умением для любого программиста.

Практическое задание: написать основную логику своего проекта. Например, для игры "Угадай число" — логику загадывания числа и сравнения с вводом пользователя.

День третий

Доработка и тестирование

- Добавление дополнительного функционала (например, меню, сохранение/загрузка);
- Тестирование программы: проверка на ошибки, обработка некорректного ввода;
- Поиск и исправление багов.

Результат занятия: ученики доведут свой проект до рабочего состояния, научатся тестировать свой код и обрабатывать возможные ошибки.

Практическое задание: реализовать дополнительный функционал и провести полное тестирование своего проекта, исправляя найденные ошибки.

День четвертый

Защита проектов и подведение итогов

- Презентация и демонстрация готовых проектов;
- Обсуждение результатов, успехов и трудностей;
- Ревью кода (code review) друг друга;
- Подведение итогов курса, рекомендации по дальнейшему обучению.

Результат занятия: ученики получают опыт публичного представления своей работы, научатся анализировать чужой код и получают мотивацию для дальнейшего развития в программировании.

Практическое задание: подготовить короткую презентацию своего проекта и выступить перед группой.