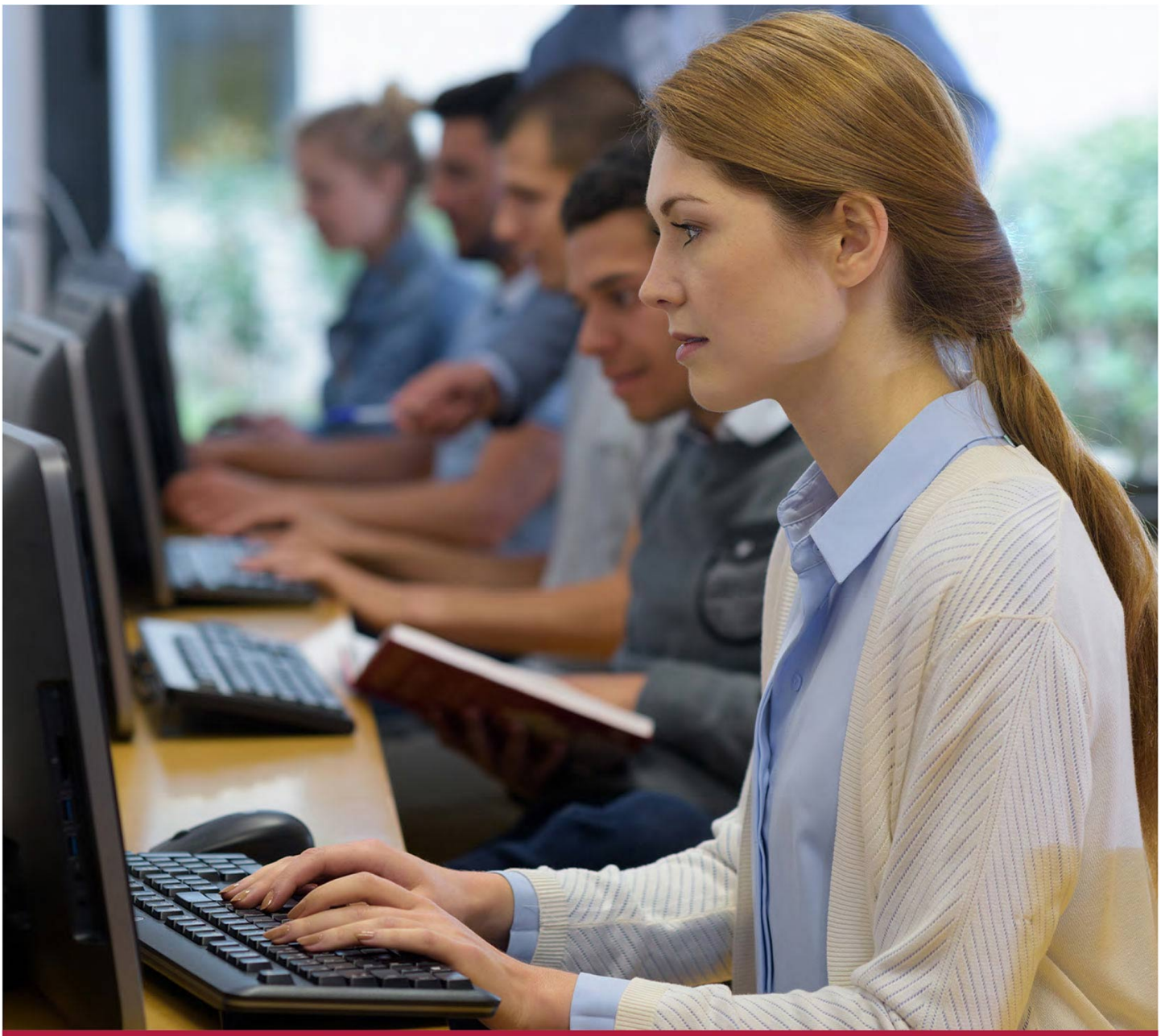
Академия АйТи
a Softline Company



Безопасное программирование Secure Coding

Код курса: SC

+7 (495) 150 96 00 | academy@academyit.ru | academyit.ru

Безопасное программирование Secure Coding

Код курса: SC

Длительность	40 ак. часов
Формат	
Разработчик курса	Академия АйТи
Тип	Учебный курс
Способ обучения	Под руководством тренера

О курсе

Безопасное программирование охватывает основные принципы безопасности и уязвимости программного обеспечения, вызванные небезопасным кодированием в рамках SDLC. На курсе рассматриваются подходы и технологии, направленные на снижение числа ошибок в коде, особенно таких, которые можно использовать как бреши в его безопасности. Также рассматриваются успешные действия agile-команд и лучшие практики от лидеров рынка (Microsoft, Google).

Подробная информация

Профиль аудитории:

- разработчики, дизайнеры и архитекторы ПО со стажем работы не менее одного года.

Предварительные требования:

- навыки программирования, желательно на C/C++;
- навыки работы без фреймворков;
- понимание концепций функционирования операционных систем.

По окончании курса слушатели смогут:

знать:

- основные понятия и терминологию в области безопасности;

уметь:

- разрабатывать, проектировать и поддерживать приложения, используя методы обеспечения безопасности в разрабатываемом коде;

владеть:

- навыками безопасного программирования и основами анализа и проектирования безопасности.

Программа курса

Модуль 1. Архитектура безопасного C++: стандарты, ревью и анализ кода

1. Корпоративные стандарты C++: внедрение Google C++ Style, автоматизация clang- format
2. Практики эффективного Code Review: чек-листы, парное программирование, внедрение Pull Request Workflows
3. Статический и динамический анализ: интеграция clang-tidy, cppcheck и AddressSanitizer в рабочий процесс
4. Документирование архитектурных решений и изменений: md, Changelog, Internal Knowledge Base
5. Инструменты командной разработки: настройка CI для проверки безопасности кода
6. Практическая сессия: Анализ и рефакторинг существующего кода с помощью автоматизированных и ручных инструментов качества

Модуль 2. Безопасное управление памятью и ресурсами в C++

1. Применение паттерна RAII и умных указателей (std::unique_ptr, std::shared_ptr, std::weak_ptr) для предотвращения утечек памяти
2. Использование современных контейнеров STL для безопасной работы с динамическими структурами данных
3. Диагностика и устранение ошибок работы с памятью: переполнение буфера, double free, use-after-free, race conditions
4. Администрирование ресурсов в многопоточной среде: thread safety, использование std::mutex, std::lock_guard
5. Валидация пользовательского ввода: предотвращение классических уязвимостей через строгую типизацию и ограничение данных
6. Практическая сессия: Поиск и исправление memory leaks, race conditions и других ошибок в лабораторном проекте

Модуль 3. Устойчивость приложений: защита данных, работа с файловой системой и обработка ошибок

1. Реализация безопасных методов сериализации и десериализации данных
2. Контроль доступа к файловой системе: защита от path traversal, безопасное открытие и сохранение файлов
3. Обработка исключений и управление ошибками: проектирование fail-safe приложений
4. Интеграция систем логирования (spdlog, Log) с учетом требований безопасности
5. Предотвращение арифметических и логических уязвимостей: integer overflow, signed/unsigned mismatch
6. Практическая сессия: Разработка и тестирование защищённых функций ввода- вывода и обработки ошибок

Модуль 4. Многопоточность, синхронизация и параллельное программирование в безопасности

1. Проектирование потокобезопасных компонентов с использованием std::thread, std::atomic, std::condition_variable
2. Профилирование и устранение Deadlock, Livelock и Race Condition с помощью ThreadSanitizer
3. Пул потоков (thread pool) и безопасный обмен данными между потоками

4. Контролируемая синхронизация доступа к общим ресурсам: Scoped Lock, Reader/Writer Locks
5. Модульное тестирование многопоточных компонентов: инструменты и стратегии
6. Практическая сессия: Аудит и устранение дефектов синхронизации в учебном многопоточном приложении

Модуль 5. Интеграция тестирования, аудит безопасности и развитие корпоративных практик

1. Модульное и интеграционное тестирование безопасности: инструменты GTest, lcov, gcov
2. Внедрение покрытия кода и контроль регресса: автоматизация проверки на уязвимости
3. Аудит безопасности существующих проектов: чек-листы, рекомендации, разбор уязвимостей CVE
4. Организация внутреннего обучения и обмена знаниями по безопасному C++
5. Планирование и документирование процесса устранения уязвимостей: incident response, багтрекеры
6. Практическая сессия: Итоговая командная работа — аудит, тестирование и защита реального или лабораторного проекта

Посмотреть расписание курса и записаться на обучение

Обращайтесь по любым вопросам
к менеджерам Академии АйТи

+7 (495) 150 96 00 | academy@academyit.ru