

# Sela.

C8

## Advanced C Programming

college@sela.co.il

03-6176666





# Advanced C Programming

C8 - Version: 3

 3 days Course

## Description:

This course is designed to expand the knowledge of C programmers by teaching some of the more advanced features of the C language, features that are often used only by experienced and knowledgeable programmers. The course material includes many examples. Since an understanding of the topics of this course is a basic need of every programmer who wants to excel in C programming, the course includes many opportunities for hands-on experience

## Intended audience:

C programmers wishing to gain a better understanding and deeper knowledge of the advanced features of the C programming language.

## Prerequisites:

Knowledge of the C programming language at a level equal to a basic C course  
Some experience in C programming

## Objectives:

Better understand the C language  
Exploit advanced features of the C language to enable efficient programming

## Topics:



## Files

- File support overview.
- ASCII and binary files.
- Working with files

## Preprocessor

- The C Preprocessor - How it works.
- #include.
- #define - Defining named constants.
- #undef.
- Predefined macros (\_FILE\_ etc.).
- Macros.
- Conditional compilation.
- Preprocessor operators #, ##, defined.
- #line, #error.
- #pragma.

## Modular Programming

- Multiple file programs
- Extern and static declaration (for variables and for functions)
- How executable file are created?
- The compiler
- The linker
- Project structure
- Object libraries

## Advanced Use of Pointers



- Arrays with negative indexes.
- Using const in pointer declarations.
- void pointers.
- Pointers to functions.
  - The library function `qsort`.
  - The library function `bsearch`
- Reallocation.
- Pointer to structures

## C Functions and the Environment

- Arguments to main
- Environment variables
  - Library functions `getenv`, `putenv` and the global variable `environ`.
- Functions with a variable number of arguments
- The library function `system`
- The library macro `assert`
- The library function `perror` and global variable `errno`
- The `atexit` function
- Recursion