born to build

42 أبوظبي ABU DHABI

ACADEMIC RESULTS FOR AMRO ELSIDDIG

I, the undersigned , Chief Executive Officer of 42 Abu Dhabi , hereby certify that:

Amro Elsiddig,

obtained the grades detailed below as of May 15, 2025.

This certificate is delivered upon request for all legal intents and purposes.

Selected in: **November 2021**
Curriculum started on: **December 19, 2021**
Curriculum ended on: **-**

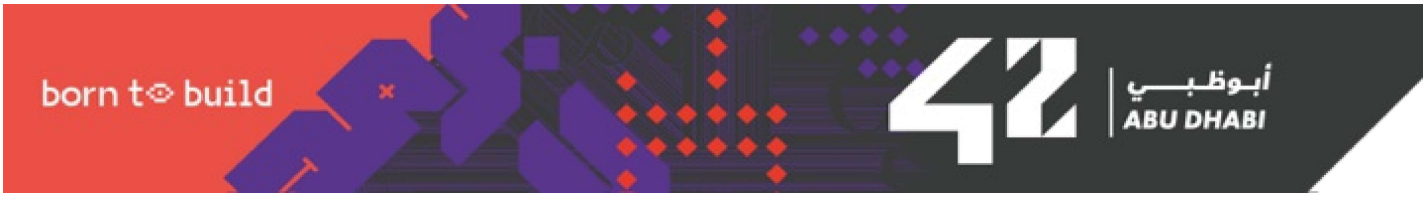
42 Abu Dhabi is part of the 42 network of ICT schools. 42 is a non-traditional educator founded in 2013 offering high-quality and scalable software engineering education to anyone who wants to learn.
It's mission to prepare the next generation for the jobs of today and tomorrow using an innovative educational model, which relies on peer-to-peer learning, project-based and hands-on approach to programming. This innovative model, allowing individual pace and path, has proven that our students become industry-ready software engineers within 2 to 5 years.

The progression of the student inside the curriculum is represented by its level, over 21.
The current level of the student is: **11.04.**

The 42 curriculum is divided into two halves: the common core and the 42 advanced part. Once students complete the first half (the common core), they have the option to either continue their journey in the 42 advanced part, or conclude their progression and become an alumni at any point during this second part.
The current situation of the student is: **in the 42 advanced part.**

Issued in Abu Dhabi on May 15, 2025 and is official when signed by the Chief Executive Officer.

Chief Executive Officer



DETAILS

Here is a description of each part of the curriculum and the current position of the student:

The Common Core

The common core of the 42 curriculum represents the minimum set of skills to be ready for a first professional experience. It provides basic and standard coding skills, as well as a fruitful range of soft skills. The delay of the CC is approximately between 1 and 2 years. The following information represent the skills developed during this part of the curriculum and the current progression of the student:

Amro Elsiddig : Common core achieved at: **100%.**

Common Core Projects

Name	Estimated workload	Result	Associated skills	Validation date
Libft	70H	Pass	Rigor, Algorithms & AI, Imperative programming	February 08, 2022
get_next_line	70H	Pass	Rigor, Algorithms & AI, Unix	February 17, 2022
ft_printf	70H	Pass	Rigor, Algorithms & AI	March 07, 2022
Born2beroot	40H	Pass with bonus	Rigor, Network & system administration	April 09, 2022
minitalk	50H	Pass	Rigor, Unix	October 23, 2022
push_swap	60H	Pass	Rigor, Algorithms & AI, Unix, Imperative programming	December 22, 2022
Exam Rank 02	0H	Pass		January 05, 2023
so_long	60H	Pass with bonus	Graphics, Imperative programming	January 06, 2023
minishell	210H	Pass	Rigor, Unix, Imperative programming	January 20, 2023
Exam Rank 03	0H	Pass		March 03, 2023
Philosophers	70H	Pass	Rigor, Unix, Imperative programming	March 12, 2023
NetPractice	50H	Pass	Rigor, Network & system administration	April 16, 2023
CPP Module 00	22H	Pass	Rigor, Imperative programming, Object-oriented programming	May 10, 2023
CPP Module 01	12H	Pass	Rigor, Imperative programming, Object-oriented programming	May 13, 2023
CPP Module 02	12H	Pass	Rigor, Imperative programming, Object-oriented programming	May 13, 2023
CPP Module 03	12H	Pass	Rigor, Imperative programming, Object-oriented programming	May 14, 2023
CPP Module 04	12H	Pass	Rigor, Imperative programming, Object-oriented programming	May 15, 2023
cub3d	280H	Pass with bonus	Rigor, Algorithms & AI, Graphics, Imperative programming	July 11, 2023
Exam Rank 04	0H	Pass		November 14, 2023
Inception	210H	Pass	Rigor, Network & system administration	November 15, 2023
CPP Module 05	25H	Pass	Rigor, Imperative programming, Object-oriented programming	December 03, 2023
CPP Module 06	25H	Pass	Rigor, Imperative programming, Object-oriented programming	December 04, 2023
CPP Module 07	25H	Pass	Rigor, Imperative programming, Object-oriented programming	December 09, 2023
ft_irc	175H	Pass with bonus	Rigor, Unix, Network & system administration, Object-oriented programming	December 09, 2023

CPP Module 08	25H	Pass	Rigor, Imperative programming, Object-oriented programming	December 22, 2023
CPP Module 09	40H	Pass	Rigor, Imperative programming, Object-oriented programming	April 05, 2024
Exam Rank 05	0H	Pass		May 23, 2024
ft_transcendence	245H	Pass with bonus	Rigor, Web, Group & interpersonal	June 12, 2024
Exam Rank 06	0H	Pass		July 18, 2024

Developed skills during the entire common core are detailed in Appendix 1 and Descriptions of each covered project are detailed in Appendix 2

The 42 Advanced

The 42 Advanced offers a choice of path among various ICT specialisations: each student can select the topic(s) she/he wants to develop and improve. This part of the curriculum also contains several professional experiences (internships, part-time jobs, ...).

- Algo & AI & Data: 1
- Cryptography & Maths: 1

Professional experience: no professional experience yet

42 Advanced Projects

Name	Estimated workload	Result	Associated skills	Validation date
Python - 0 - Starting	7H	Failed	Rigor, Algorithms & AI, Object-oriented programming	July 25, 2024
ft_linear_regression	70H	Pass with bonus	Rigor, Algorithms & AI, DB & Data	March 12, 2025
Python for Data Science	35H	in progress	Rigor, Algorithms & AI, Object-oriented programming	-

Internship and professional experiences				
Company name	**Duration**	**Validation**	**Skills**	**Validation date**
–				

SPECIAL

A student can eventually benefit from special programs or projects valuable for their personal skill set, and thus included in their curriculum. They are mentioned here:

Name	Equivalent workload
–	

APPENDIX 1

Developed skills during the entire common core :

- ***Algorithms & AI:*** Standards algorithms on standards structures: searching, sorting, insertion, deletion, balance, on: arrays, linked lists, trees. State machine and asynchronous management.

- ***Graphics:*** Image management, RGB structure of an image, manipulating areas, drawing into an image, interacting with the window management system and getting user events and inputs from keyboard and mouse, programming with callbacks and event loop.

- ***Group & interpersonal:*** Collaboration, relationships and group management situations, including different kinds of interactions between people (friendly, tensions ...)

- ***Imperative programming:*** Basics of coding in C : the C syntax, variable, loops, conditional branches, functions, recursivity, instructions, calculus and expressions, comparisons operators, standard and advanced types, strings processing, structures, includes and libraries, memory allocation and release, linked lists, trees, the C standard library

- ***Network & system administration:*** Basics of computer networking : IP addresses, subnets, default routing, local network structure, host to host connectivity to network services; Basics of system administration : operating system installation with Linux, setting up security, access, users, storage, installing network services like mail, dns, web server, ...

- ***Object-oriented programming:*** Object programming principles in C++, classes, namespaces, constructors and destructors, memory management in C++, inheritance, abstraction, overloading, templates, standard C++ library types and tools

- ***Rigor:*** The need to fulfill administrative and technical constraints. The need for a wide and deep testing process to eliminate failure.

- ***System programming:*** Classic Unix system interactions : system calls, filesystem access and management, process creation, execution, management; inter-process communications : pipes and signals; device management and ioctl, terminal capabilities; network communication : TCP & UDP sockets, DNS resolution, endianness

- ***Web:*** The client-server architecture involved in the web, role and actions of the web server, role and actions of the web browser; The HTTP protocol; Web technologies involved : HTML, CSS, Javascript, images and videos; Backend language and framework for dynamic websites: one among php, ruby, python, go, javascript, Rails, Symfony, Django, Node, ... ; MVC model; users web services : web sessions, authentification, cookies, search, caddie, backoffice configuration, ... ; Basics of user experience, user interface, and design.

APPENDIX 2

Description of each project:

- **Libft**: Covers standard C functions, including memory manipulation, string handling, and linked lists.
- **ft_printf**: Implements the printf function in C, focusing on output formatting and type conversion.
- **get_next_line**: Reads a line from a file descriptor, managing dynamic memory and multiple descriptors.
- **Born2beroot**: Involves setting up a Linux VM, managing users, permissions, and remote administration via SSH.
- **minitalk**: Facilitates inter-process communication using Unix signals, handling pipes and message sending.
- **pipex**: Simulates terminal pipe behavior, executing sequential commands and redirecting outputs.
- **so_long**: Focuses on 2D game development with MiniLibX, featuring sprite animations and path validation.
- **Fdf**: Renders 3D landscapes from altitude data using line-drawing algorithms and geometric transformations.
- **fract-ol**: Generates interactive fractals like Mandelbrot and Julia, allowing user exploration of configurations.
- **push_swap**: Covers sorting algorithms through stack implementations.
- **Philosophers**: Explores concurrent programming, addressing deadlocks and multithreading with mutexes and semaphores.
- **minishell**: Replicates Unix shell behavior, executing commands and managing environment variables and redirections.
- **CPP Module 00**: Introduces object-oriented programming in C++, focusing on classes, namespaces, and member functions.
- **CPP Module 01**: Covers memory allocation and file stream handling in C++.
- **CPP Module 02**: Focuses on polymorphism and operator overloading in C++.
- **CPP Module 03**: Discusses inheritance and access specifiers in C++.
- **CPP Module 04**: Explores subtype polymorphism and abstract classes in C++.
- **NetPractice**: Covers networking basics like IP addressing and routing, configuring small networks.
- **cub3d**: Creates a 3D maze game using raycasting techniques, inspired by Wolfenstein 3D.
- **miniRT**: Develops a ray tracing engine for 3D rendering with realistic shadows and reflections.
- **CPP Module 05**: Introduces exception handling and flow control in C++.
- **CPP Module 06**: Explores different casting types in C++.

- **CPP Module 07**: Covers templates and their usage in C++.
- **CPP Module 08**: Discusses templated containers and algorithms using the STL.
- **CPP Module 09**: Explores generic algorithms and advanced data structures in C++.
- **ft_irc**: Implements a chat server based on the IRC protocol, handling multiple connections.
- **webserv**: Builds an HTTP server from scratch, focusing on network protocols and web services.
- **Inception**: Configures a complete web development environment using Docker containers.
- **ft_transcendence**: Develops a full-stack single-page web application (SPA) for real-time multiplayer Pong.