

ACADEMIC RESULTS FOR YUNUS EMRE SARAÇ

I, the undersigned Osman Melih Gökçe, Director of 42 Türkiye, hereby certify that:

Yunus Emre Saraç, born on August 02, 2005 in Kocaeli (Turkey)

obtained the grades detailed below as of January 06, 2026.

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

Selected in: September 2023

Curriculum started on: October 09, 2023

Curriculum ended on: -

Founded in 2013, 42 is a worldwide network of ICT schools. We are a non-traditional educator offering high-quality and scalable software engineering education to anyone who wants to learn.

It is our mission to prepare the next generation for the jobs of today and tomorrow. We do so using an innovative educational model, which relies on peer-to-peer learning, project-based and hands-on approach to programming. Our 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: 11.17.

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.

See details below.

Made in Kocaeli, on January 06, 2026

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:

Yunus Emre Saraç : Common core achieved at: 100%.

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, authentication, cookies, search, caddie, backoffice configuration, ... ; Basics of user experience, user interface, and design.

Details of each validated project in appendix 1.

The 42 Advanced Part

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, ...).

No projects completed yet

Professional experience: no professional experience yet

Details of the validated projects in appendix 2.

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

Projects covered during the common core:

Name	Estimated workload	Result	Associated skills	Validation date
Libft	70H	Pass with bonus	Rigor, Imperative programming, Algorithms & AI	October 28, 2023
ft_printf	55H	Pass	Rigor, Algorithms & AI	November 04, 2023
get_next_line	55H	Pass with bonus	Rigor, Unix, Algorithms & AI	November 22, 2023
Born2beroot	50H	Pass	Rigor, Network & system administration	December 12, 2023
Exam Rank 02	0H	Pass		January 05, 2024
minitalk	50H	Pass with bonus	Rigor, Unix	January 08, 2024
push_swap	50H	Pass	Rigor, Imperative programming, Unix, Algorithms & AI	February 29, 2024
so_long	60H	Pass	Imperative programming, Graphics	March 20, 2024
Exam Rank 03	0H	Pass		March 22, 2024
Philosophers	70H	Pass	Rigor, Imperative programming, Unix	May 21, 2024
minishell	210H	Pass with bonus	Rigor, Imperative programming, Unix	July 23, 2024
CPP Module 00	22H	Pass	Object-oriented programming, Rigor, Imperative programming	September 02, 2024
Exam Rank 04	0H	Pass		October 18, 2024
CPP Module 01	12H	Pass	Object-oriented programming, Rigor, Imperative programming	January 12, 2025
CPP Module 02	12H	Pass	Object-oriented programming, Rigor, Imperative programming	January 13, 2025
CPP Module 03	12H	Pass	Object-oriented programming, Rigor, Imperative programming	January 18, 2025
CPP Module 04	12H	Pass	Object-oriented programming, Rigor, Imperative programming	January 19, 2025
NetPractice	50H	Pass	Rigor, Network & system administration	January 24, 2025

cub3d	280H	Pass with bonus	Rigor, Imperative programming, Algorithms & AI, Graphics	February 09, 2025
ft_irc	175H	Pass with bonus	Object-oriented programming, Rigor, Unix, Network & system administration	May 28, 2025
CPP Module 05	25H	Pass	Object-oriented programming, Rigor, Imperative programming	June 18, 2025
CPP Module 06	25H	Pass	Object-oriented programming, Rigor, Imperative programming	June 20, 2025
CPP Module 07	25H	Pass	Object-oriented programming, Rigor, Imperative programming	June 25, 2025
CPP Module 08	25H	Pass	Object-oriented programming, Rigor, Imperative programming	June 26, 2025
CPP Module 09	40H	Pass	Object-oriented programming, Rigor, Imperative programming	June 28, 2025
Inception	150H	Pass	Rigor, Network & system administration	July 07, 2025
Exam Rank 05	0H	Pass		August 22, 2025
Exam Rank 06	0H	Pass		November 14, 2025
ft_transcendence	245H	Pass with bonus	Rigor, Web, Group & interpersonal	November 27, 2025

APPENDIX 2

Projects covered during the 42 advanced:

Name	Estimated workload	Result	Associated skills	Validation date
—				

Internship and professional experiences				
Company name	Duration	Validation	Skills	Validation date
—				

APPENDIX 3

Description of each covered project:

Name	Description
Libft	This project is your very first project as a student at 42. You will need to recode a few functions from the C standard library, as well as some other utility functions that you will use throughout your whole curriculum.
ft_printf	This project is pretty straightforward, you have to recode printf. You will learn what is and how to implement variadic functions. Once you validate it, you will reuse this function in your future projects.
get_next_line	Whether it's a file, stdin, or even later a network connection, you'll always need a way to read content line by line. It's time to start working on this function, which will be essential for your future projects.
Born2beroot	This project aims to introduce you to the wonderful world of virtualization.
Exam Rank 02	□ Exam Rank 02 □
minitalk	The purpose of this project is to code a small data exchange program using UNIX signals. It is an introductory project for the bigger UNIX projects that will appear later on in the cursus.
push_swap	This project involves sorting data on a stack, with a limited set of instructions, and using the smallest number of moves. To make this happen, you will have to manipulate various sorting algorithms and choose the most appropriate solution(s) for optimized data sorting.
so_long	This project is a small 2D game with minilibx. You'll learn about textures, sprites and tiles.
Exam Rank 03	□ Exam Rank 03 □
Philosophers	This project aims to teach concurrent programming, focusing on multithreading and multiprocessing.
minishell	The objective of this project is for you to create a simple shell.

CPP Module 00	This first module of C++ is designed to help you understand the specificities of the language when compared to C. Time to dive into Object-Oriented Programming!
Exam Rank 04	EXAM RANK 04
CPP Module 01	This module is designed to help you understand memory allocation, references, pointers to members, and the usage of the switch statement in C++.
CPP Module 02	This module is designed to help you understand ad-hoc polymorphism, function overloading, and orthodox canonical classes in C++.
CPP Module 03	This module is designed to help you understand inheritance in C++.
CPP Module 04	This module is designed to help you understand subtype polymorphism, abstract classes, and interfaces in C++.
NetPractice	NetPractice is a hands-on networking project featuring 10 progressive levels that teach essential computer networking fundamentals. Through interactive problem-solving, you'll master TCP/IP addressing, subnet masks, default gateways, routing, and OSI layers by troubleshooting and configuring non-functioning network diagrams. This browser-based training provides practical experience in network administration, preparing you for real-world system administration and networking challenges.
cub3d	This project is inspired by the world-famous eponymous 90's game, which was the first FPS ever. It will enable you to explore ray-casting. Your goal will be to make a dynamic view inside a maze, in which you'll have to find your way.
ft_irc	Create your own IRC server in C++98, compatible with a standard IRC client for the required features.
CPP Module 05	This module is designed to help you understand try/catch and exceptions in C++.
CPP Module 06	This module is designed to help you understand the different types of casting in C++.
CPP Module 07	This module is designed to help you understand templates in C++.
CPP Module 08	This module is designed to help you understand templated containers, iterators, and algorithms in C++.
CPP Module 09	This module is designed to help you understand containers in C++.
Inception	Broaden your system administration skills by working with Docker. In this project, you'll set up a complete infrastructure using Docker Compose, creating and managing multiple containerized services including NGINX with SSL/TLS, WordPress with php-fpm, and MariaDB. You'll gain hands-on experience with containerization, networking, volume management, and secure web service deployment within your own personal virtual machine.
Exam Rank 05	
Exam Rank 06	
ft_transcendence	Design, develop, and organize a full-stack web application with complete creative freedom. Choose your project concept, select from a wide range of technical modules, and make key architectural decisions. This highly flexible project allows you to explore modern web development while demonstrating your technical skills and creativity through a modular approach.