42 Berlin
Harzer Straße 39
12059 Berlin
GERMANY

ACADEMIC RESULTS FOR AARON PAZ MARTINEZ

I, the undersigned, Daniel Hadley, Pedagogy Lead of 42 Berlin, hereby certify that:

Aaron Paz Martinez, born on August 25, 1986 in Santiago de Compostela (Spain)

obtained the grades detailed below as of December 14, 2025.

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

Selected in: **July 2023**
Curriculum started on: **April 22, 2024**
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 in the curriculum is represented by their level. The highest level is 21.
The current level of the student is: **5.12.**

The 42 curriculum is divided into two halves: the Core Curriculum and the 42 Advanced Curriculum. Once students complete the first half (the Core Curriculum), they have the option to either continue their journey in the 42 Advanced Curriculum or stop their studies and become an alumni at any point during this second part.
The current status of the student is: **in the Common Core.**
See details below.

Made in Berlin, on December 14, 2025

DETAILS

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

The Core Curriculum

The Core Curriculum of 42 comprises of the minimum set of skills to be ready for a first professional experience. It provides fundamental and diverse coding skills, as well as a broad range of soft skills. The duration of the Core Curriculum is between 1 and 2 years depending on the student's pace. The following information represents the skills developed during this part of the curriculum and the current progression of the student:

Aaron Paz Martinez : Core Curriculum achieved at: **57%.**

Developed skills during the entire Core Curriculum:

- ***Algorithms & AI:*** Standard algorithms on standard structures: searching, sorting, insertion, deletion, balance, on: arrays, linked lists, trees. Machine state 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.

- ***Imperative programming:*** Basics of coding in C : the C syntax, variables, loops, conditional branches, functions, recursivity, instructions, calculus and expressions, comparison operators, standard and advanced types, string 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 servers, ...

- ***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 comprehensive testing process to reduce points of 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.

Details of each validated project in appendix 1.

The 42 Advanced Curriculum

The 42 Advanced Curriculum offers a choice of paths among various ICT specialisations: each student can select the topic(s) they want 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 also complete special programs or projects valuable for their skill set, and thus include them in their curriculum. They are mentioned here:

Name	Equivalent workload
–	

APPENDIX 1

Projects covered during the Core Curriculum:

Name	Estimated workload	Result	Associated skills	Validation date
Libft	70H	Pass with bonus	Rigor, Imperative programming, Algorithms & AI	May 20, 2024
ft_printf	55H	Pass	Rigor, Algorithms & AI	June 16, 2024
Born2beroot	50H	Pass	Rigor, Network & system administration	July 02, 2024
get_next_line	55H	Pass	Rigor, Algorithms & AI, Unix	July 22, 2024
Exam Rank 02	0H	Pass		August 30, 2024
push_swap	50H	Pass	Rigor, Imperative programming, Algorithms & AI, Unix	November 13, 2024
pipex	50H	Pass	Imperative programming, Unix	November 28, 2024
FdF	60H	Pass with bonus	Rigor, Imperative programming, Algorithms & AI, Graphics	December 16, 2024
Exam Rank 03	0H	Pass		March 14, 2025
minishell	210H	Pass	Rigor, Imperative programming, Unix	May 21, 2025
Philosophers	70H	Pass	Rigor, Imperative programming, Unix	July 06, 2025
Exam Rank 04	0H	Pass		August 12, 2025
NetPractice	50H	Pass	Rigor, Network & system administration	October 02, 2025
cub3d	280H	Pass with bonus	Rigor, Imperative programming, Algorithms & AI, Graphics	November 04, 2025
CPP Module 00	22H	Pass	Object-oriented programming, Rigor, Imperative programming	November 27, 2025
CPP Module 01	12H	Pass	Object-oriented programming, Rigor, Imperative programming	December 02, 2025
CPP Module 02	12H	Pass	Object-oriented programming, Rigor, Imperative programming	December 02, 2025
CPP Module 03	12H	Pass	Object-oriented programming, Rigor, Imperative programming	December 08, 2025
CPP Module 04	12H	Pass	Object-oriented programming, Rigor, Imperative programming	December 14, 2025

APPENDIX 2

Projects covered during the 42 Advanced Curriculum:

Name	Estimated workload	Result	Associated skills	Validation date
–				

Internships and professional experiences				
Company name	Duration	Validation	Skills	Validation date
–				

APPENDIX 3

Description of each completed 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.
Born2beroot	This project aims to introduce you to the wonderful world of virtualization.
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.
Exam Rank 02	This project will evaluate your abilities and knowledge about programming.
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.
pipex	This project aims to deepen your understanding of the two concepts that you already know: Redirections and Pipes. It is an introductory project for the bigger UNIX projects that will appear later on in the cursus.
FdF	All programs that you wrote until now were executed in text mode on your terminal. Now, let’s discover something more exciting: how to open a graphics window and draw inside? To start your journey in graphic programming, FdF offers to represent “iron wire” meshing in 3D.
Exam Rank 03	This project will evaluate your abilities and knowledge about programming.
minishell	The objective of this project is for you to create a simple shell.
Philosophers	This project aims to teach concurrent programming, focusing on multithreading and multiprocessing.
Exam Rank 04	This project will evaluate your abilities and knowledge about programming.
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.
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!
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++.