



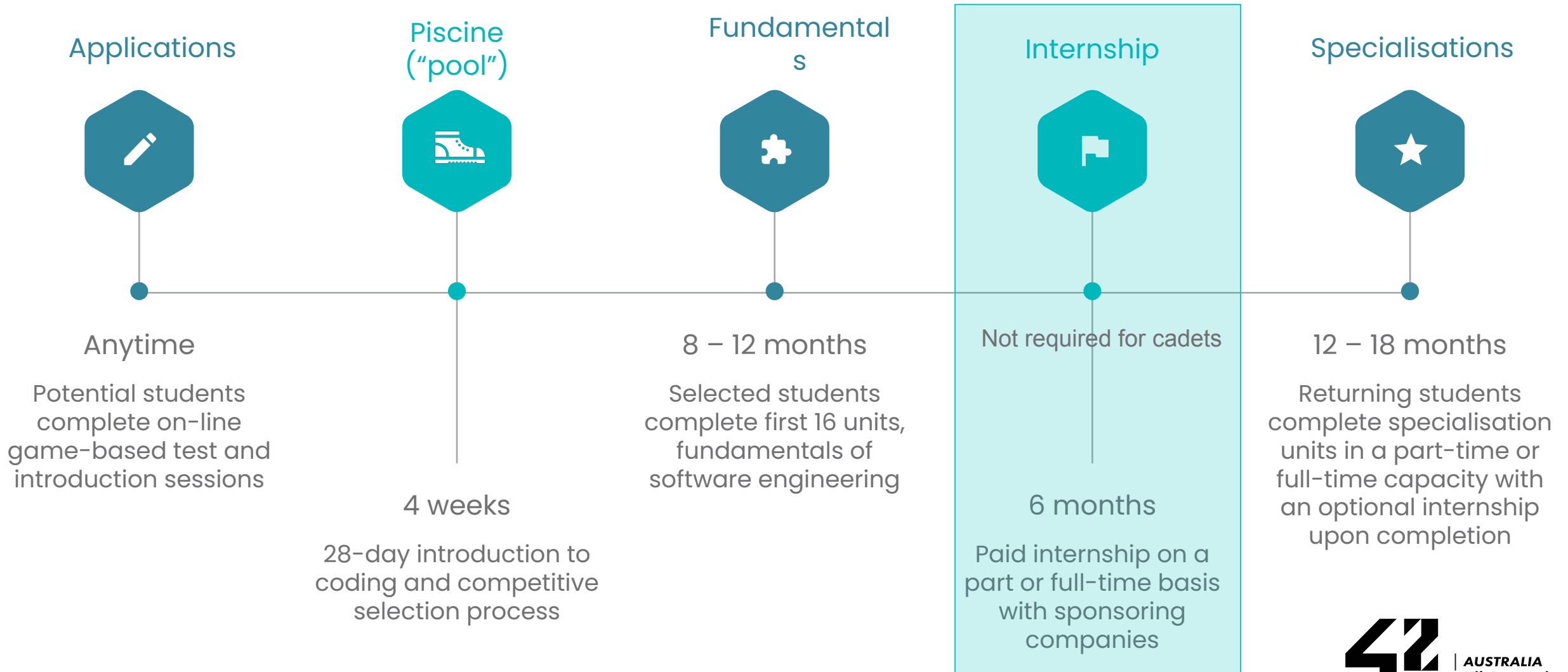
42 Curriculum

Updated: 1st August 2022



The 42 student journey

Our students learn at their own pace with their peers, learning how to learn. This means that our partners have access to talent and interns all year round.

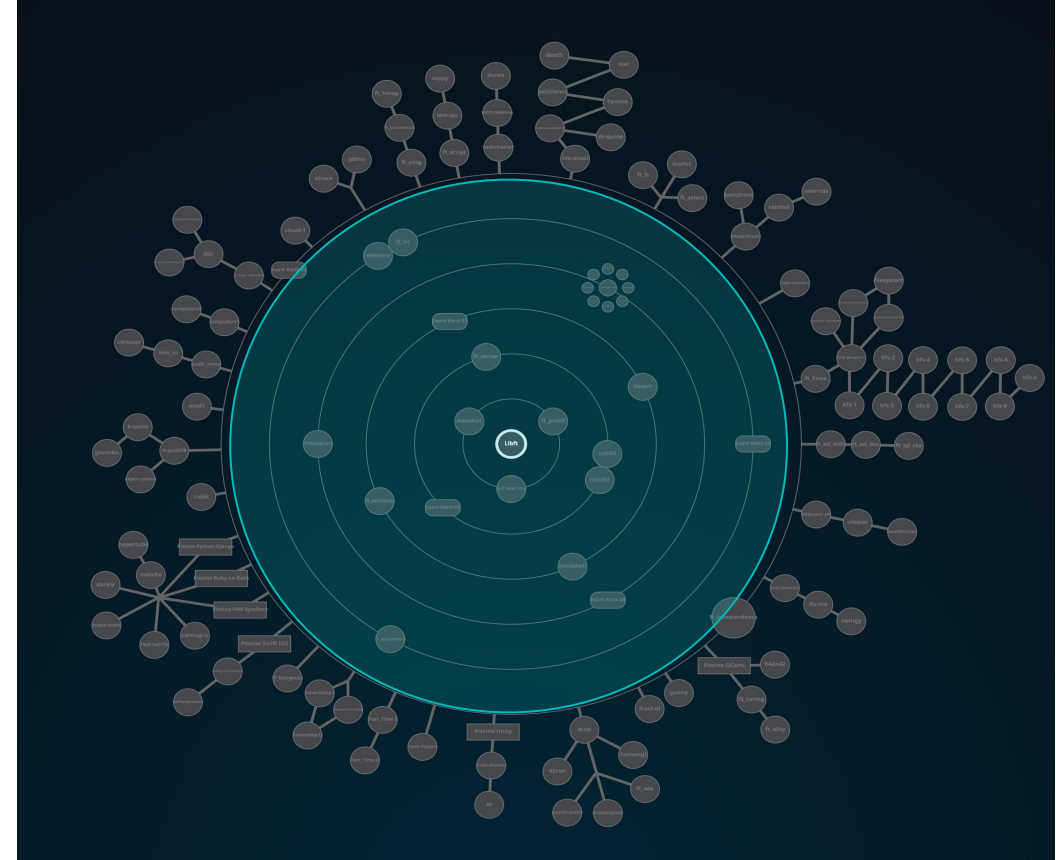


Curriculum overview

The 42 program is divided into 2 parts.

The **first stage** is known as the inner circle which focuses on our students building a strong foundation in programming, networking and systems. Once completed the students do a 6 month internship with our hiring partners (This does not apply to cadets as they will be working on this as part of their cadetship).

The **second stage** is known as the outer circle and is completed by students after their internship has been completed (optional), often in a part-time capacity as they have secured employment. This part of the curriculum focuses on specialisations and is available to 42 students at any time.



Each circle shows a project within the 42 game based curriculum.
Green shows the inner circle
The outer circle shows the specialisations

The curricula

While technical skills are the foundation of learning at 42 it is the critical skills gained that sets our 42 graduates apart.

Technical skills

- Programming
 - Imperative
 - Functional
 - Object-oriented
- Algorithms
- Artificial intelligence
- Graphics
- Technology integration
- System administration
- Networking
- Security
- Data & databases
- Parallel computing

Critical skills gained

- Communication
- Collaboration
- Adaptation
- Problem solving
- Autonomy
- Personal organization
- Self-learning
- Innovation
- Resilience
- Critical thinking
- Analytical
- Learning



42

ADELAIDE

Inner circle

First 16 units
(part 1 of the curriculum)



Inner circle overview

Within the first part of the curriculum the students are trained in software engineering and build a solid foundation.

Overview

Students are trained as software engineers with the right skills and mindset to be productive in the workplace from day 1. They quickly adapt to different frameworks and are used to working according to strict standards in terms of quality and style. Students have experience with intensive collaboration and communication in teams.

Form of education

- Practical work (16 projects)
- Exams (5)
- Internship (4-6 months)

Hours:

- Average 2720 hours (1920 practical work + 800 internship) – varies between students

Inner circle elements:

- C
- C++
- System administration, servers and networks
- Web development

Free choice in language and framework for +5 projects – including C, C++, JavaScript, PHP, and Java, with the option to use others like Ruby, Python, Kotlin and more. They will also be required to develop Web-based Applications using HTML, CSS, JavaScript and PHP. Various implementations of Databases feature throughout the program to ensure candidates are well acquainted with data and data linking.

Element 1 – Programming in C

Overview

Our Students write complex programs in C and create and implement standard functions. As they progress through the projects they must then use them in subsequent projects, developing an in-depth understanding of the programming language and the underlying concepts. These include; data types, memory management, data structures (linked lists, binary trees, abstract syntax tree, recursion, variadic arguments, lexer, parser, graphic rendering (raycasting/raytracing), unix filesystem, multithreading (mutex/semaphore).

Student Technical Learning outcomes

- Convert a problem into a workable coding solution
- Write code in C that meets strict standards, style and structure requirements
- Write and implement standard library functions (string functions, printf)
- Write complex programs with minimal tools (graphics engine, shell)
- Write applications that use multithreading
- Implement standard library features in Assembly code

Projects	Description	Type
Libft	Write a library consisting of 43 functions that can be used in all subsequent projects. Features can be divided into three categories: 1: libc features like memmove, strcat, atoi, calloc and strdup. 2: additional features like substr, strjoin, itoa. 3: list features like lstnew, lstadd_back, lstmap (bonus).	Solo
Get_next_line	Write a function that returns the next line of a file descriptor when calling. A static variable must be used to hold a buffer in it.	solo
Ft_printf	Develop their own implementation of the standard printf function. Contains the conversions: 'cspdiuxX%' and flags: '-O.*'. Must work if the 'real' printf and variadic arguments use it.	solo
Cub3d	Develop a mini 3D game using ray casting. The goal is to create a dynamic view in a maze, in which to move.	solo
miniRT	Develop a basic raytracer. Can generate an image using the raytracing protocol.	solo
Minishell	Develop a shell-based bash-based implementation. Must include: pipes, environment variables, signals, redirections, quotes, multiple commands, CD, pwd, export, echo, unset, env, exit.	group
Philosophers	An introduction to multithreading and working with processes that use the same memory at the same time. Topics: threads, mutex, semaphore, processes	solo
libasm	Develop a library in Assembly consisting of 11 functions.	solo

Exams	Description
Rank 02	Write a simplified implementation of get_next_line or printf
Rank 03	Develop a simple graphics rendering (mini_paint or micro_paint)
Rank 04	Develop a minishell: shell implementation with pipes and CD
Rank 06	Write a simple web server (mini_serv)

Element 2 – Programming in C++

Overview

Our students learn and become proficient in object oriented programming (OOP) and develop in-depth understanding of different techniques and how to apply them in one of the larger projects at the end of the first part of the curriculum.

Topics covered include: Encapsulation, Polymorphism, Inheritance, Classes, Abstract classes, Templates, STL, List, Vector, Map, Stack, Queue.

Student Technical Learning outcomes

- Know how to develop object-oriented code (classes, methods, inheritance, and interfaces)
- Have insight into designing programs aligning with good design structure
- Write complex programs in C++, utilising web servers, IRC client/server.
- Write their own implementations of standard C++ containers
- Coding with an object oriented web framework

Projects	Description	Type
CPP Module 01 t/m 08	Extended introduction to C++ and object oriented programming (OOP). Topics covered are Class, Inheritance, Abstract classes, Templates, STL	solo
Ft_containers	Develop implementation of the standard C++ Containers List, Vector, Map, Stack, and Queue.	solo
Ft_irc	Write an Internet Relay Chat (IRC) server	solo
Webserv	Develop an HTTP server in C++ according to the rfc 7230-7235 (http 1.1) standard.	Group

Exams	Description
Rank 05	Write 10 classes that require all the gained knowledge about OOP in C++.

Element 3 – Systems, servers and networks

Overview

Our students learn the basics of networking and systems administration, which includes topics such as IP addressing, netmasks, TCP, DHCP, DNS, routing and ports.

They will learn how to deploy a web server using ‘Docker’ technology running multiple services including Wordpress, phpMyAdmin, and a SQL database and also how to set up a multi service cluster of those services using Kubernetes for managing and deployment.

Project	Description	Type
NetPractice	An introduction to networking through multiple choice questions. Topics: IP, netmask, TCP, DHCP, DNS, routing, ports.	solo
Inception	Set up a web server using Docker. Topics: Nginx, MySql, phpMyAdmin, WordPress, SSL	solo
Webserv	Writing a HTTP server, test with browser, ports and hosts, execute CGI. cookies, session management	solo

Student Technical Learning outcomes

- Understanding the basic concepts of System Administration
- Scripting to automate tasks
- Set up a complex cluster of containers using Kubernetes by virtualising a network
- Develop coding best practices to reduce security breaches
- Understanding ip address plans, subnet, routing
- Operating system installation and management, using Linux
- Local networks and host connections
- Set up and maintain network services (mail, web, dns, ...)
- Create and deploy simple containers and virtual machines

Element 4 – Web development

Overview

Our students learn to build an extensive online platform.
The covers include: HTML, CSS, BackboneJS, Rails , PostgreSQL.

Student Technical Learning outcomes

- Develop usable fullstack website
- Use javascript libraries and frameworks
- Develop single page applications (login, Chat, Game, Groups, administration panel)
- Use of relational PostgreSQL database
- Creating full featured web-site using a classic framework (Django, Rails, Symfony)
- Approach of UX/UI/Design
- Authentication, session handling and backend configuration

Project	Description	Type
Ft_transcendence	Annual changing web development project*. The programming languages and frameworks to be used can change each year. A comprehensive online application must be written (both backend and front-end).	Team 3-5

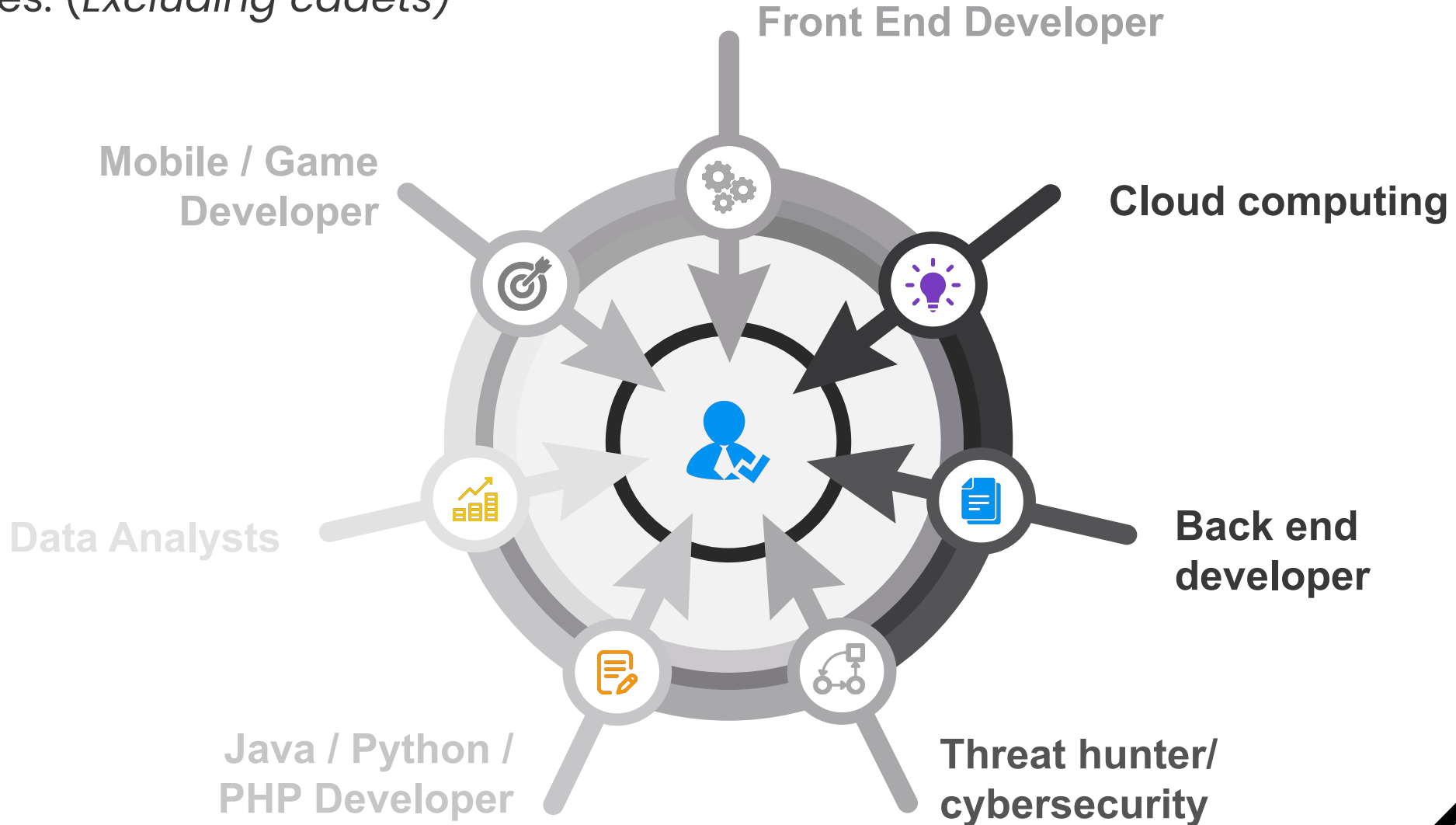
** 2020 Project: Develop an intranet for a school with a programming language and framework unknown to them. The choice of programming language and framework is up to the student provided they are applications that he or she has not previously worked with. See the full project description for illustration here.*

Student organisational learning outcomes

- Learn how to problem solve independently as well as in teams, to come up with effective and robust solutions
- Solve complex programming issues in new and unknown environments
- Apply different programming languages and frameworks to develop and implement software solutions across a variety of domains
- Have the ability to communicate, receive and provide feedback about their own code and explain its operation to others with various programming knowledge
- Have the ability to make mistakes and demonstrably learn from them to ultimately implement the most optimal solution
- Contribute to large software development processes by working in teams, breaking the issue into sub-problems, writing clear and well documented code, and reviewing each other's work
- Independently collect, analyse, assess and use information
 - Combine information from different (types of) sources
 - Dealing with and filtering large amounts of information
- Robust error and input handling
 - Learning from mistakes and continuously improving solution
 - Be accurate in evaluating one's own work and the work of others
 - Taking responsibility for the process (time management) and results
 - Ask others for help and offer help
 - Collaborate with people from different backgrounds and levels
 - Solve complex problems in a group and work them out in code
- Give and receive feedback through code reviews
- Business analysis
- Leadership

Internship and career pathways

After the inner circle (first 16 units) the students complete a 6 month internship with companies. (*Excluding cadets*)



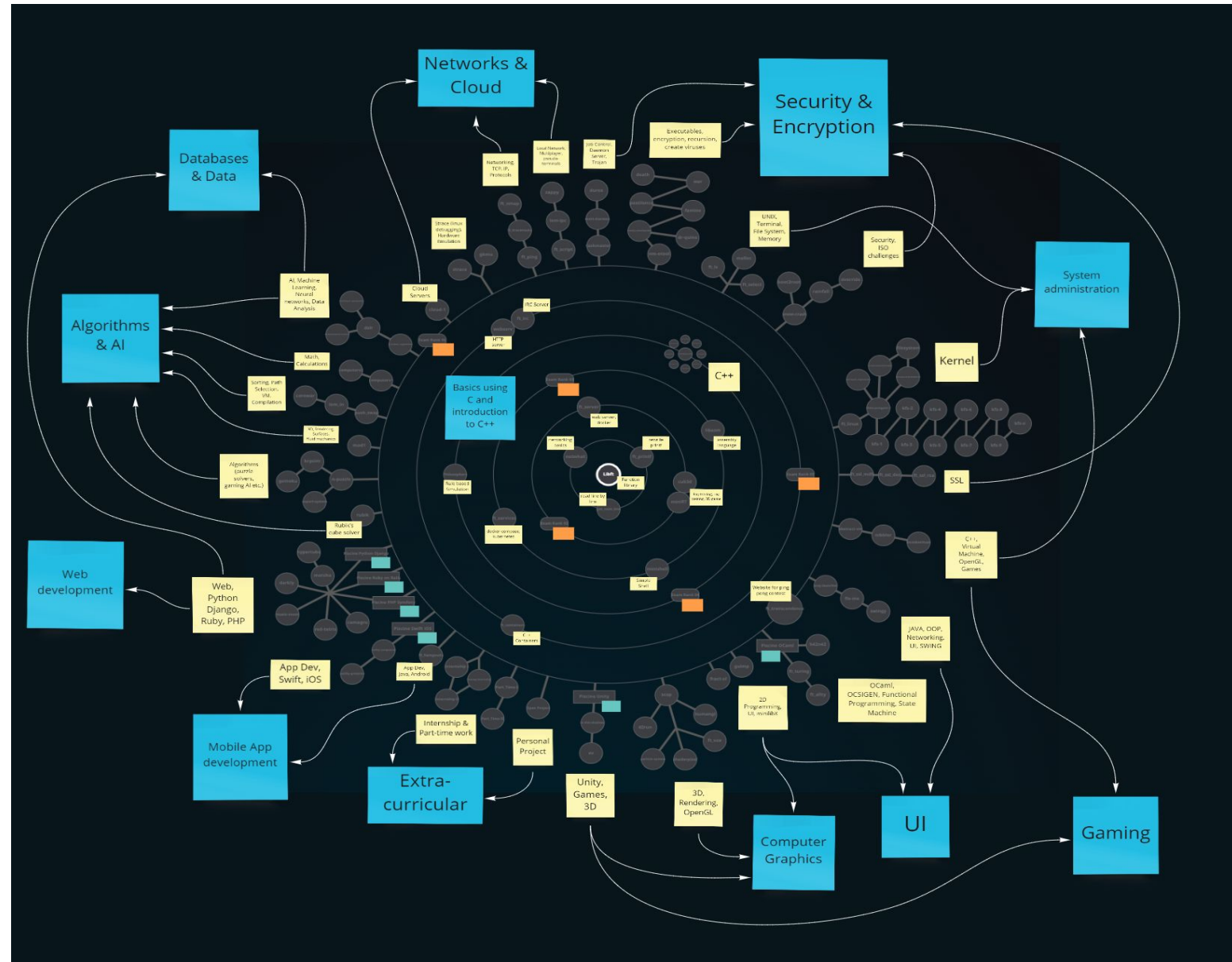


Outer circle

Specialisations



Map of specialisations



Specialisation

Gaming and Creative



Creative – gaming and film pathways

Year 1 project examples

The first year students at 42 learn the foundations of coding and can start their journey towards a careers in film / gaming, including introductions to 3D graphics, ray casting and ray tracing

cub3d

This project is inspired by the world-famous eponymous 90's game. It will enable students to explore ray-casting. The goal will be to make a dynamic view inside a maze, in there they'll have to find The way Create the first RayCaster with miniLibX

Rigor, use of C, use of basic algorithms. As a graphic design project, cub3D will enables improved skills in these areas: windows, colours, events, fill shapes, etc

CODE: C **PROJECT:** Individual

MiniRT

This project is an introduction to the beautiful world of Raytracing. Once completed students will be able to render simple Computer-Generated-Images and they will never be afraid of implementing mathematical formulas again.

CODE: C **PROJECT:** Group or Individual

FT Transcendence

Build a website for the mighty pong contest. the website will help user run pong tournament and play against each other. There will be an admin view, chat with moderators, real time multiplayer online games.

CODE: Rails **PROJECT:** Group

Creative – Gaming and graphics pathways

Year 2 (Outer Circle) project examples

C++
Virtual Machine
Open GL
Games

Abstract VM

The creation of a virtual machine capable of executing a code in basic pseudo-ASM. Students will thus use a stack for basic arithmetic operations, such as the dc command.

CODE: C++ **PROJECT:** Individual

Nibbler

This project in C++ will make students recode a Snake where students can switch the interfaces at run time.

CODE: C++ **PROJECT:** Group or Individual

Bomberman

Here students will take advantage of its simplicity to make students first big C++ project with a professional look. students will recode a version of Bomberman, in C++ and with OpenGL, with the aim of making the game as successful as possible.

CODE: C++ **PROJECT:** Group

3D
Rendering
Open GL

scop

students will become acquainted with the main concepts of 3D. Create a small program that will show a 3D object conceived with a modelling program like Blender. The 3D object is stored in a .obj file. students will write a parser to obtain the requested rendering.

CODE: Open **PROJECT:** Individual

HumanGL

This intermediate open GL project is an introduction to hierarchy modelling. students will learn to use matrices in order to link various parts of a humanoid model, and animate them.

CODE: Open **PROJECT:** Group

42 run

Students goal is to create a small program that will present an endless run (within the school walls) in 3D, while using the codes of temple run/temple run 2 gameplay. Need to show elements including a set that shows movement, obstacles, a set with cool perspective.

CODE: Open **PROJECT:** Individual

Shaderpixel

Display 3D fractals, procedural worlds, clouds all within an exhibition window created to showcase students creations. CGI through the use of shaders, and more precisely OpenGL fragment shaders. students will also have to learn how to optimize students displays and light effects

CODE: Open **PROJECT:** Group

Particle-system

In this project, students will implement a particle system using the OpenCL API and its kernel programming language OpenCL C to give life to students particle system. students will also render them using OpenGL.

CODE: Open **PROJECT:** Individual

Ft-vox

Introduction to the voxel engine, inspired by minecraft, in which students must create a randomly generated world.

CODE: Open **PROJECT:** Group

Creative – Gaming and graphics pathways cont.

Year 2 (Outer Circle) project examples

Unity Games 3D

2 week – BOOTCAMP “Piscine” Unity

This set of projects starts with a bootcamp as a way of an introduction to Unity. This is an intensive 2 weeks of peer based learning split in many small projects completed each day, and with each day a new notion on the language/paradigm/technology students chose to work on is introduced.

CODE: Open **PROJECT:** Individual and Group
released daily`

Projects include:

Day 1 – Assets, GameObject, Behaviour,
Input, Transform

Day 4 – Advanced inputs and 2D GUI

Day 7 – Staging, shaders, lights, 3D sound

Day 10 – Putting it all together

Day 2 – 2D physics, Tags, Layers and
Scenes

Day 5 – PlayerPrefs and Coroutines

Day 8 – Navmesh, 3D Physics, 3D GUI

Group Project 1

Day 3 – Audio, Animation and communication between
scripts

Day 6 – Terrain, camera management, 3D Physics Terrain

Day 9 – Animations

Group Project 2

In-the-shadows

In this project, students will code a whole game using Unity, students will develop a game close to Shadowmatic.

CODE: Open **PROJECT:** Group

XV


Build an application with Unity allowing its users to create and animate a virtual industrial environment.

CODE: Open **PROJECT:** Group

Student examples – creative

A student profile who has taken the creative pathway


Linked in profile –
<https://www.linkedin.com/in/hadrien-estela-a5427a95/>




Hadrian Estela
Admiral[?]
Promo November 2013

✉ hadria@student.42.fr
in linkedin / hadrien-estela-a5427a95


ACHIEVEMENTS[?]



I FOUND THE ANSWER
Validate level 21 and be ready to face the outside world!



RIGOROUS BASTERD 4
Validate 21 projects in a row (pool days included).



404 - SLEEP NOT FOUND
To be logged in 24 hours a day. (to work, ofc!)

SKILLS[?]

UNIX ● ● ● ●

GRAPHICS ● ● ● ●

EXPERTISE[?]

CPP UNITY VS#

FIRST INTERNSHIP CONTRACT[?]

BUSINESS
microDON
170 Losserand Street
75014, Paris
France

TRAINEEESHIP

Connected object project - Lead design & web - Hardware & software prototyping
170 Losserand Street
75014, Paris
France
from 01/11/2014 to 31/12/2014

REVIEWS

Mid-term evaluation by the company
Hadrien got 100
Final evaluation by the company
Hadrien got 100

FINAL INTERNSHIP CONTRACT[?]

BUSINESS
PinPin Team
91 rue du Faubourg Saint-Honoré
75008, Paris
France

TRAINEEESHIP

Unity Developer
79 Rue du Faubourg Poissonnière
75009, Paris
France

REVIEWS

Mid-term evaluation by the company
Hadrien got 100
Final evaluation by the company
Hadrien got 100



Specialisation

Web & Mobile App
Development



Web Development Pathways

Before students complete more advanced projects they participate in one or more **2 week intensive bootcamps** focusing on a specific language to learn key components of web development.

Bootcamp 1 Python

This set of projects starts with a bootcamp as a way of an introduction to Web Development using the chosen language as the foundation. This is intensive peer based learning split into many small projects completed each day, and a new notion on the language or paradigm introduced each

Bootcamp 2 Ruby on Rails

- HTTP, HTML, CSS and JS integration
- Libraries
- Authentication and user management
- Contextualisation
- Servers and production

Bootcamp 3 Symfomy (PHP)

- Syntactics and semantics
- Media and interface design
- Inheritance and exceptions
- Sites (static and dynamic)

Advanced Projects

*Language / code is
free choice based
on above
bootcamp selected,
unless indicated*

Matcha

Students create a dating web app with the following features; registration & sign in, user profile, browsing, research, chat, notifications, integration Facebook, Google etc.

PROJECT: Group

Music room

A complete mobile solution focused on music and user experience, created in collaboration with Deezer. The student creates a complete mobile app of a collaborative playlist using the SDK and API of Deezer. On the menu, mobile native development, back-end development and creation of API REST.

PROJECT: Group

Hypertube

Discover MVC frameworks. The student will learn how to build an MVC application, in the language of their choice, to create a video streaming site downloaded via the BitTorrent protocols.

PROJECT: Group

Red Tetris

The student will create an advanced asynchronous network protocol to develop a networked multi player Tetris style game. Using functional programming, algorithmic animation will be graphically displayed in HTML.

CODE: JavaScript **PROJECT:** Group

FIXME

Discover computer security in relation to the web through dissecting a vulnerable website, become aware about security in a web application and problems associated with simple development errors, both from a programming and design perspective

PROJECT: Individual or Group

Camagru

Using PHP the student will build a small Instagram-like website allowing users to create and share a photomontage. The application will be built from scratch, delivering basic functionalities of user driven websites.

PROJECT: Individual

App Development Pathways

Year 2 (Outer Circle) project examples

App Development Swift Java / iOS

2 week – iOS Swift App Development Bootcamp (Piscine)

This set of projects starts with a bootcamp as a way of an introduction to Advanced App Development using iOS as the foundation. This is an intensive 2 weeks of peer based learning split in many small projects completed each day, and with each day a new notion on the language/paradigm/technology the student choses to work on is introduced.

Projects include:

Day 1 – Introduction to Xcode – develop calculator app
Day 4 – Photos, multithread, warnings, scroll view, zoom
Day 7 – TapGesture, dynamic behaviour, gestures, CoreMotion
Day 10 – Putting it all together

Day 2 – Introduction to Swift, using classic card games
Day 5 – Tweet, API, requests, research
Day 8 – Pods (reuse functionality)
Group Project 1

Day 3 – Tables and lists, navigation, basics
Day 6 – Tab, MapKit, Segment, ControlBar, Geo-tracking, locations
Day 9 – Pods, data modelling, class articles
Group Project 2

CODE: Swift /XCODE **PROJECT::** Individual and Group released daily`

Swiftly Companion

The student will build an iOS or Android app that will retrieve the information of 42 students, using the 42 API. Students will use Xcode and Android Studio IDE's.

CODE: Swift/Kotlin/Java **PROJECT:** Individual

Swiftly Protein

This project will give the student the opportunity to discover Scene Kit, an Apple framework to create 3D scenes. Students will make an application that models ligands (molecules) in 3D. This deepens knowledge of Swift frameworks such as Touch ID, sharing under iOS, gestures as well as SearchBar, Touch ID and social Sharing mobile API

CODE: Swift /XCODE **PROJECT:** Group

T- Hangouts

The student will create a contact management mobile app on Google's Android platform. They will have to understand how an Android app functions, how Android manages their application and how to use the SD.

CODE: Java **PROJECT:** Individual

Creative – UI Pathways

Year 2 Project Examples

Java
OOP
Networking
UI
Swing

Abstract VM

First project of the Java projects arc. Implementation of a simple Java program according to a given class diagram written in UML.

CODE: Java **PROJECT:** Individual

Swingy

Develop GUI applications with the SWING framework, in order to create an RPG game.

CODE: Java **PROJECT:** Individual

FIXME

Write a program that simulates stock exchanges and deals with trading algorithms, with networking and socket implementations.

CODE: Java **PROJECT:** Individual

2D Programming
UI
MiniLibX

GUIMP

Create a graphic interface library. The student will have to prove it works correctly with a small 2D image software. This library must as complete and modular as possible, the goal being to re-use it in future projects, whether for projects in the graphic branch or other projects of their training.

CODE: Open **PROJECT:** Individual or Group

Fract-ol

Discover 2D programming and the psychedelic universe of fractals in this project, using MiniLibX.

CODE: C **PROJECT:** Individual

In year one and two we can also co-create specific projects / curricula for students in collaboration with our partners and sponsors



| ADELAIDE

Specialisation

Security & Encryption

Our students learn:

- Coding and defending against viruses
- Using and coding software that exploit a security breach
- Network oriented attacks : man in the middle, sniffing, IP & ARP spoofing,
- System and administration projects that need a high level of security, with firewalling, backup, DMZ, NAT.
- Explore the various cryptographic algorithms : DES, RSA, AES, MD5, ... and create their own implementation



Security Pathways

Year 2 (Outer Circle) project examples

SSL

ft_ssl_md5

An introduction to cryptographic hashing algorithms, This project is the gateway to the encryption branch. The student will recode part of the OpenSSL program, specifically the MD5 and SHA-256 hashing algorithms.

PROJECT: Individual

ft_ssl [base64] [des]

The student will learn about symmetric encoding and encryption. This project is a continuation of the previous encryption project. They will recode part of the OpenSSL program, specifically BASE64, DES-ECB and DES-CBC.

PROJECT: Individual

ft_ssl [genrsa] [rsa] [rsautl]

This project is a continuation of the previous encryption project. The student will recode part of the OpenSSL program, specifically GENRSA, RSA, and RSAUTL. By the end of this they should be intimately familiar with the workings of asymmetric key cryptosystems.

PROJECT: Individual

Unix Terminal File System Memory

ft_ls

Learn the filesystem inside out, and how files and directories are sorted. In short this project will get the student to recode the "ls" command and learn how to interact with the file system.

PROJECT: Individual or Group (2)

ft_select

As a Unix learning project, the student will code a small user interface program using termcaps which will allow them to pick from a list of choices and return it to the shell.

PROJECT: Individual

malloc

This project is about implementing a dynamic memory allocation mechanism, through writing a dynamic allocation memory management library.

PROJECT: Individual

Job Control Daemon Server Trojan

Taskmaster

The goal of this project is to make a Unix job control daemon, with features similar to supervisor. The students will learn to manage Job control in Unix using multiprocessing.

PROJECT: Group

matt-daemon

A Unix project to create a daemon of type server. This server will listen on a given port, store and interpret a list of command messages.

PROJECT: Individual or Group (2)

durex

students goal in this project is to code a simple trojan horse.

PROJECT: Individual or Group (2)

Security Pathways

Year 2 (Outer Circle) project examples

Security
ISO

Snow-crash

Introduction to computer security. The student will learn to spot different techniques used to exploit software weaknesses. Using several languages (ASM/perl/php), they will learn to differentiate between unknown programs and simple programming errors.
PROJECT: Individual or Group (2)

Boot 2 root

This project is an introduction to penetrating a computer system
PROJECT: Group

Rainfall

This project aims to further the student's knowledge in the world of elf-like binary exploitation in an i386 system. Reverse engineering and reconstructing code to understand and detect faults.
PROJECT: Group

Override

Override follows on from the Rainfall project, in which the student will have to search for faults present in the protected binaries, and re-build these binaries depending on their behaviour.
PROJECT: Individual or Group

Executables
Encryption
Recursion
Create Viruses

nm-otool

Go deep into the format of MacOS executables and understand how the kernel launches binaries by re-writing the nm and otool commands. This is a must for all those who want to work in security. More generally, this project is an opening on UNIX system culture.
PROJECT: Individual

Famine

Introduction to the wonderful world of viruses, trojans and other anti virus software. The goal here is to put the students' skills handling binary files to make their first virus.
PROJECT: Group

woody-woodpacker

A logical follow-up project on nm/otools that is designed to modify the headers of a ELF64 file. The goal here is to add a chunk of code and hide a non-stripped part of a file.
PROJECT: Individual or Group (2)

pestilence

Second virus project. The student will reuse the Famine base to add a hiding method for their code.
PROJECT: Group

war

This project will reuse the pestilence project, the student will make the binary modify its signature at runtime. They will code a "polymorphic" virus.
PROJECT: Group

dr-quine

This algorithm project will get the student acquainted with auto-replication problems and encounter the Kleene recursion theorem.
CODE: Open **PROJECT:** Group

death

This project will build on the student's Famine, Pestilence and War projects to create a real "metamorphic" virus.
CODE: Open **PROJECT:** Individual



| ADELAIDE

Specialisation

System Administration

- Through complex system administration and networking projects, face and solve cohabitation problems between a large variety of usual software and services
- Infrastructure and administration in the cloud
- Design, manage and maintain a complex network
- Able to put in place CI/CD to fulfil a Devops position
- Coding a device driver inside a Unix Kernel
- Create micro kernel with minimal
- Explore binary file format, replicating a classic unix tool
- Re-coding the system standard memory allocator
- Mixing network programming, computer graphics and AI, create a full online game using a client-server architecture.
- Creating debugging tools in a Linux environment.



System Admin Pathway

Year 2 (Outer Circle) project examples

UNIX

Terminal
File System
Memory

ft_ls

Learn the filesystem inside out, and how files and directories are sorted. In short this project will get the student to recode the "ls" command and learn how to interact with the file system.

PROJECT: Individual or Group (2)

malloc

This project is about implementing a dynamic memory allocation mechanism, through writing a dynamic allocation memory management library.

PROJECT: Individual

ft_select

As a Unix learning project, the student will code a small user interface program using termcaps which will allow them to pick from a list of choices and return it to the shell.

PROJECT: Individual

Kernel (path 1)

ft_linux

The first project of the Kernel branch! This is a simple LFS so that the student can build their own distribution which will be used in the next projects.

PROJECT: Individual

little penguin

The start of a series of challenges inspired by Eudypytula. The student will get acquainted with many points of Kernel development by compiling a custom kernel, build and using a kernel module and learning how linux drivers work.

PROJECT: Individual

userspace_digressions

The student will learn about the init system, run levels, systemd case through designing a boot ready program, understanding userspace by creating their own userspace init binary.

PROJECT: Group

drivers-and-interrupts

Learn how to connect a driver to students kernel using a keyboard driver. The student will learn how to work with interrupt requests, understand interrupt key functions and associate a miscellaneous device with a driver.

PROJECT: Individual

process-and-memory

An introduction to syscalls and memory management within linux kernel. The student will write a working syscall and explore user and kernel memory.

PROJECT: Individual

filesystem

The student will create their own filesystem for their linux kernel, they will learn about filesystems (superblocks, inodes, work with rights, links and interact with other filesystems.

PROJECT: Individual or Group (2)

System Admin Pathways

Year 2 (Outer Circle) project examples

Kernel (path 2)

ft_linux

The first project of the Kernel branch! This is a simple LFS so that the student can build their own distribution which will be used in the next projects.

PROJECT:: Individual

drivers-and-interrupts

Learn how to connect a driver to students kernel using a keyboard driver. The student will learn how to work with interrupt requests, understand interrupt key functions and associate a miscellaneous device with a driver.

PROJECT: Individual

ft_ls

Learn the filesystem inside out, and how files and directories are sorted. In short this project will get the student to recode the "ls" command and learn how to interact with the file system.

PROJECT: Individual or Group (2)

little penguin

The start of a series of challenges inspired by Eudypatula. The student will get acquainted with many points of Kernel development by compiling a custom kernel, build and using a kernel module and learning how linux drivers work.

PROJECT: Individual

process-and-memory

An introduction to syscalls and memory management within linux kernel. The student will write a working syscall and explore user and kernel memory.

PROJECT: Individual

malloc

This project is about implementing a dynamic memory allocation mechanism, through writing a dynamic allocation memory management library.

PROJECT: Individual

kfs-1

By the end of this project the student will have a kernel they can boot, a basic kernel library and a basic kernel.

PROJECT: Individual or Group (2)

filesystem

The student will create their own filesystem for their linux kernel, they will learn about filesystems (superblocks, inodes, work with rights, links and interact with other filesystems.

PROJECT: Individual or Group (2)

ft_select

As a Unix learning project, the student will code a small user interface program using termcaps which will allow them to pick from a list of choices and return it to the shell.

PROJECT: Individual



| ADELAIDE

Specialisation

Data Analysis, Databases and AI

- Solving realistic problems using genetic algorithms
- Using neural networks for machine learning projects
- Discover and use tools dedicated to machine learning : python libraries, TensorFlow, ..
- Using classic game theory algorithms in small games, like A*
- Design of a databases using UML
- Coding a data visualisation system
- Advanced database structures/systems: Nosql, lake.



Data, AI and algorithms

Year 2 Project Examples

AI
Machine Learning
Neural Networks
Data Analytics

Cloud-1

This project is about cloud discovery with auto-scaling, load-balancing, and CDN. This project is done in a simulated work environment with a budget, decision making and project management requirements in partnership with Amazon Web Services

ft_linear_regression

This project will be the student's first steps into AI and Machine Learning. They will start with a simple, basic machine learning algorithm, creating a program that predicts the price of a car by using a linear function train with a gradient descent algorithm.

Datascience X Logistic Regression

Discover Data Science in the projects where students re-constitute Poudlard's Sorting Hat, covering data analysis, data visualisation and logistic regression. The student will learn how to read a data set, visualize it in different ways, to select and clean unnecessary information from students data. They will train a logistic regression that will solve classification problem

multilayer-perceptron

This project is an introduction to Machine learning through artificial neural networks, with the implementation of a multilayer perceptron.

total-perspective-vortex

Use python tools to process large dimension data, and create a man-machine interface thanks to brainwaves. This subject aims to create a brain computer interface based on electroencephalographic data (EEG data) with the help of machine learning algorithms.

These subjects should be completed in conjunction with the Python bootcamp and web development core modules

Data, AI and algorithms

Year 2 Project Examples

Math calculations

Sorting

Path selection

VM

Compilation

3D Rendering

Surfaces

Fluid Mechanics

Algorithms

computorv1

This project is about Mathematics Basics and Polynomial Equations. It aims to make students code a simple equation solving program. It will take polynomial equations into account. These equations will only require exponents. No complex function. The program will have to display its solution(s).

computorv2

Dive deeper into mathematics for coding and analytics including, polynomial equations, Imaginary number calculations, Matrix calculations and Mathematical functions. Through creating a calculator in command line, which will integrate students computer V1 as well as functions for matrix calculus, function resolution, calculation with complex numbers, etc.

Mod-1

Learn about Bezier surfaces, advanced algorithms and fluid mechanics. This simulation graphic project will have students represent a surface in 3D, on which water will flow. Empirical or scientific (with many equations), recreate a wave, a tsunami or maybe just rain on students surface.

push_swap

This project involves sorting algorithms and cattery concept and handling elements. It involves sorting data on a stack, with a limited set of instructions, and the smallest number of moves. The student will manipulate various sorting algorithms and choose the most appropriate solution(s) for optimized data sorting.

lem_in

This project moves the student deeper into algorithms and gets them acquainted with graph traversal algorithms: The program developed will have to intelligently select paths and precise movements..

corewar

In this project students will learn, compilation, simplistic virtual machine, simplistic assembly type language and visual rendering. It involves creating a virtual arena, and having simplistic-language programs fight each other. The student will get acquainted with VM conception and compilation problems of an assembly language in bytecode.

project_x

Learn more in depth about Genetic algorithm, Simplex algorithm and Production Optimization through designing a program that will read a file describing processes, analyse the whole, and propose a worthy solution.