

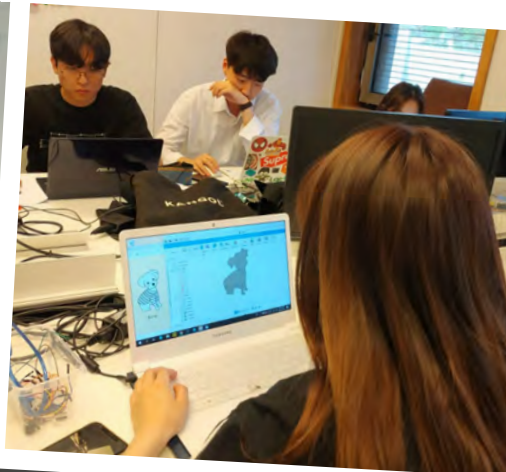
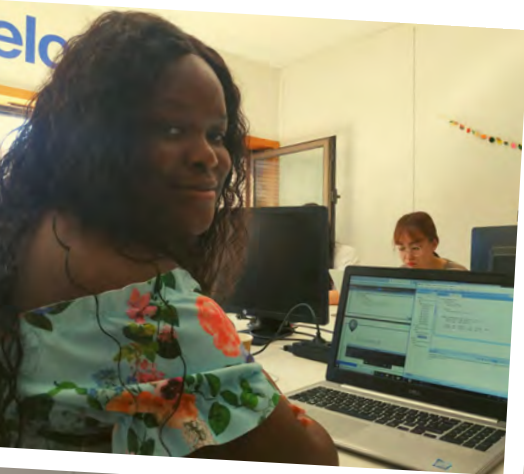
# EPITECH.

JOIN US AT

EPITECH - SCHOOL OF IT & INNOVATION

**Courses offered in English (2022-2023)**

FALL SEMESTER 2022/ACADEMIC YEAR





# COURSE CATALOGUE 2022/2023

*Fall semester/Academic year*

*September 2022 to July 2023*

## Epitech first-year program

### Program aim:

Introducing students to the basics of programming with C language as a main tool.

### Learning outcomes:

- Strong knowledge in C
- Strong basis of programming
- Basics of Graphic programming

### Prerequisites:

- Basic knowledge of programming
- English language proficiency equivalent to B2

### Courses list:

<u>Code</u>	<u>Course</u>	<u>Credit (ECTS)</u>	<u>Semester</u>
B-CPE-100	Unix & C Lab Seminar I	4	Fall
B-CPE-101	Unix & C Lab Seminar II	3	Fall
B-CPE-110	Elementary Programming in C Part I	4	Fall
B-MUL-100	C Graphical Programming	5	Fall
B-MAT-100	Mathematics	2	Fall
B-PSU-100	Unix System Programming Part I	2	Fall
B-PSU-101	Unix System Programming Part II	3	Fall
B-NSA-100	Network and System Administration	2	Fall
B-INN-000	Guided Project – Innovation Hub	6	Fall/Spring
B-MET-100	Professional Competences	3	Fall/Spring



M-FLE-000	French Language course I	2	
Fall			
B-SEC-200	Introduction to Cyber security	2	Spring
B-CPE- 200	Elementary Programming in C – Part I	5	Spring
B-CPE- 201	Elementary Programming in C – Part II	4	Spring
B-PSU-200	Unix System Programming	4	Spring
B-PSU-210	Shell Programming	5	Spring
B-MUL-200	Graphical Programming	9	Spring
B-AIA-200	Introduction to AI	3	Spring
B-WEB-200	Introduction to Web Development	2	Spring
B-DOP-200	Introduction to DevOps	2	Spring
B-MAT-200	Mathematics	2	Spring
M-FLE-000	French Language course II	2	Spring

## Courses description

### Fall semester

#### [B-CPE-100] - Unix & C Lab Seminar I

The C Pool is the core module in the Epitech curriculum. **Unix & C Lab Seminar I** is the first part.

It will enable the students to implement the initial concepts acquired, and especially will enable them to state their own hypotheses and to run tests in order to find the solutions. In addition to the projects, you will work on “Rush”- a type of mini-project to be completed in small groups and in a limited amount of time.

#### Skills to be acquired

- UNIX shell
- C language
- Basics of UNIX system use
- Fundamental elements of C language
- Makefiles
- Data structures

#### Teaching methods

All the materials are available online. Academic mentors help you to go through the module. **Unix & C Lab Seminar I** represent a key module at Epitech. In addition to the knowledge and skills you will acquire, you will get additional skills in time management/team work as if you would be in a company. The module sets up the basic knowledge. Throughout, the students create hypotheses, runs tests while collaborating and sharing ideas between each other.



Credit value

4 ECTS

Assessments

Online project submission

Project Example

**Name:** Rush

**Subject:** Basic principles of C language

**Aim:** Train student to overcome the stress and pressure inherent to crunch time (important amount of work in a short period of time. For this project, it is equivalent to 2 days)

## [B-CPE-101] - Unix & C Lab Seminar II

The C Pool is the foundation of the Epitech curriculum. **Unix & C Lab Seminar II** is the second part.

It will enable the students to implement the initial concepts acquired and put it into practice in the exercise of Bistromatic.

Skills to be acquired

- C Language
- Project management

Teaching methods

Following the first part of the module, the students work on a stumper – project done by pairs of students with tight deadline and without external help-. This module is important since it works as a logical continuity with the first part of the module.

It aims at giving the students the tools to get started at Epitech.

Credit value

3 ECTS

Assessment

Online project submission

Project example

**Name:** Bistromatic

**Subject:** Basic principles of C language

**Aim:** Test students on C language skills acquired during the Pool in a small as well as complex project. The students will learn how to think efficiently and how to implement the well-known algorithms to solve the given problem.

## [B-CPE-110] - Elementary Programming in C (Part I)



**Elementary Programming in C (Part I)** covers the common programming aspects such as simple data structures (arrays, linked lists, stacks) in different scenarios (data sorting, identifying holes in grid maps).

#### Skills to be acquired

- Be capable of showing accuracy: reading a subject correctly, respecting a coding style to have a clear and logical code, following the rendering's instructions...
- Demonstrate the ability to correctly employ the language: syntax and simple data structures
- Demonstrate the ability to solve simple problems by using simple algorithms
- Make at least one functional rendering

#### Teaching methods

The students work on two projects for a total duration of 6 weeks of work. Each project is evaluated individually with automated tests which give the students and the academic team the information on the completion of each project.

Both projects are to be completed on their own.

#### Credit value

4 ECTS

#### Assessment

Online project submission

#### Project example

**Name:** pushswap

**Subject:** Creating an algorithm to order a list of numbers using a limited set of operations

**Aim:** Manipulate data structures and explore existing sort algorithms.

## [B-MUL-100] - C Graphical Programming

**C Graphical Programming** consists in creating images/animations and scenes management using algorithms and graphical resources.

The unit aims at introducing the notion to the students and, in a medium term, to prepare the students to the main issues of the MyRPG, which is the main project of the module at the end of the year.

The two important axes of such a project are:

- Graphical special effects generation, 2D/3D rendering, collision management, entity management, etc. being part of the mathematical component.
- Gameplay, animations with sprites, virtual world coherence, lifespan of the game, etc. everything enhancing the "game" aspect.

#### Skills to be acquired

##### Gaming projects

- Resources, Event, and Windows management.
- Persistent data storing (e.g. highscore, or progression)
- Simple starting and pause menus
- Implementation of gameplay mechanics
- Implementation of simple game physics mechanics
- Common visual effects (e.g. parallax)



### Mathematics

- Frames and windows management
- Drawing (simple and complex) shapes with a pixel drawing function.
- Implementation of simple tricks to render complex visual effects.
- Entity management
- Implementation of simple collision algorithm (hitbox)
- Optimization techniques to render and manage large numbers of entities.

### Teaching methods

The students work on two projects for 9 weeks of work. The two important axes of the project are:

- Graphical special effects generation, 2D/3D rendering, collision management, entity management
- Gameplay, animation with sprites, virtual world coherence, lifespan of the game, ...

Before the project, the student can choose between the gaming or the mathematical projects.

### Credit value

5 ECTS

### Assessment

Online project submission.

### Project examples

**Name:** MyScreensaver

**Subject:** Animations based on light or any visual effects and using particles.

**Name:** MyHunter

**Subject:** Game with a fixed camera, in which the player can shoot in some moving elements.

## [B-MAT-100] - Mathematics

**Mathematics** focuses on the mathematical tools. The unit is a complementary introduction to programming (reviewing some of the Pool's elements) and is also an introduction to scientific programming.

### Skills to be acquired

- Linear algebra: vector analysis, matrix calculus
- Geometry: geometric transformations and coordinate systems
- Nonlinear equation solving

### Teaching methods

The students work on five mini-projects of two weeks each for a total duration of 10 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project. The students will work on projects which includes several skills to be acquired. It gives a solid basics of vector calculus and matrix calculus.

The students work on each project alone or in pairs.



Credit value

2 ECTS

Assessment

Online project submission.

Project example

**Name:** 102architect

**Subject:** Compute transformation matrix that have been created using homogeneous coordinates.

**Aim:** Test the student ability to implement the mathematical concept of transformation matrix and use them in a computer software.

## [B-PSU-100] – Unix System Programming (Part I)

**Unix System Programming (Part I)** covers all the fundamental elements of UNIX programming.

In this first part, we cover the advanced management of terminals and argument lists.

Skills to be acquired

- Understand and interact with the UNIX operating system
- Get a good level in C programming
- Learning how to make system calls
- Solve algorithmic problems
- Understand logic process

Teaching methods

The students work on two mini-projects for a total duration of 5 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

Credit value

2 ECTS

Assessment

Online project submission

Project example

**Name:** my\_ls

**Subject:** Create your own version of the 'ls' binary to display list of files.

**Aim:** Know how to programmatically interact with the filesystem.

## [B-PSU-101] – Unix System Programming (Part II)

**Unix System Programming (Part II)** covers all the fundamental elements of UNIX programming.

In this second part, we cover the use of the filesystem and the creation of processes.



### Skills to be acquired

- Understand FileSystem
- Execute binary files
- Processor management and interrupt.

### Teaching methods

The students work on two mini-projects for a total duration of 5 weeks of work. Each project is evaluated individually with automated tests giving the student and the pedagogical team information about the completion of each project.

### Credit Value

3 ECTS

### Assessment

Online project submission

### Project example

**Name:** navy

**Subject:** Re-create the famous Battleship games using signals to communicate between processes

**Aim:** Inter-process communication using signals

## [B-NSA-100] – Network and System Administration

**Network and System Administration** teaches the students to master your Unix exploitation system.

### Skills to be acquired

- Install and configure an exploitation system on the command line
- Manage users and their rights and permission
- Configure a graphic environment

### Teaching methods

The students work on one project - my web for 4 weeks.

They must configure the two operating systems on two different virtual machines.

The students are evaluated on the installation and the configuration of the three operating systems – Arch Linux, Debian, Ubuntu server. They are also evaluated on the installation and configuration of a web server (Apache, using virtual hosts).

### Credit value

2 ECTS

### Assessment

Online project submission

### Project example

**Name:** my\_web

**Subject:** Install a small environment of VM with various services

**Aim:** Know how to make a dual boot installation and basic network configurations





## [B-INN-000] Guided Project- Innovation Hub

Guided project teaches the students to practice all their knowledge and skills acquired in a business environment. They will be advised and supervised by the Innovation Center of Epitech.

The module starts with an introductory session with all the international students.

### Skills to be acquired

- Project management
- Ideation and brainstorming
- Prototyping
- Documentation
- Communication and persuasion skills

### Teaching methods

Guided project with monthly follow-ups supervised by the Epitech Innovation Center Team.

In parallel, the students must attend several activities of his choice in topics of IT offered by the Innovation Center to validate the module.

### Credit value

6 ECTS

### Assessment

Project submission

## [B-MET-100] – Professional Competencies

**Professional competencies** is mainly composed of review sessions.

The reviews allow the students to present what they have completed for the project. They also must explain the reasons why they choose to follow this path and how they get to this result.

The code reviews allow the students to share programming experience and get feedbacks from their peers.

### Skills to be acquired

Each review will be evaluated on 4 elements:

- Quality Assurance (unit tests, functional tests, no code defects, ...)
- Methodology and Organization (frequencies of commit, use of branches or external tools, group or personal organization, work methods, ...)
- Involvement (going that extra mile than what the project demands)
- Quality of oral presentation



### Teaching methods

The reviews are open to any other students. An academic member evaluate it based on several criteria. Code reviews are not evaluated. The evaluator helps the students on commenting each other code.

### Credit value

3 ECTS including:

- 2 ECTS based on the participation and quality of the reviews.
- 1 ECTS based on the participation of at least two code reviews

### Assessment

Oral presentation

## [M-FLE-000] French I Language Course

The **French** module teaches the students French (writing, listening, speaking) from the beginning level

### Credit value

2 ECTS

### Assessments

Class attendance and final exam

## Spring semester ( only for students having completed Fall semester)

## [B-SEC-200] – Introduction to Cyber Security

**Introduction to Cyber Security** focuses on the basics of cyber security. It introduce the fields of forensics using Capture The Flag style challenges.

### Skills to be acquired

- Usage of tools to help exploit web application breaches
- Understanding and exploiting breaches such as XSS, CSRF, include...
- Understanding some unix concept and how to exploit them (ssh, ftp, nmap)

### Teaching methods

The unit is made of a Capture The Flag tournament in which the students must complete security challenges.

### Credit value

2 ECTS

### Assessment

The evaluation is based on the number of points and challenges validated on the CTF platform and a theoretical evaluation.



## [B-CPE-200] B2 – Elementary Programming in C (Part I)

**Elementary Programming in C (Part I)** covers common programming aspects such as algorithms and data structures in different scenarios (pathfinding, graphs, ...).

### Skills to be acquired

- Be capable of showing accuracy: reading a subject correctly, respecting a coding style to have a clear and logical code, following the rendering's instructions...
- Demonstrate the ability to correctly employ the language: syntax and simple data structures
- Demonstrate the ability to solve simple problems by using simple algorithms
- Make at least one functional rendering

### Teaching methods

The students work on two projects for a total duration of 7 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

A project is to be made with a group of up to 2 students. The second one requires a group of 3 to 4 students.

### Credit value

5 ECTS

### Assessment

Online project submission

### Project example

**Name:** Dante's star

**Subject:** Generate and resolve labyrinths

**Aim:** Pathfinding algorithms

## [B-CPE-201] – Elementary Programming in C (Part II)

**Elementary Programming in C (Part II)** corresponds to the second part of the module, Elementary Programming in C (Part I). The second part covers advanced notions linked to some programming aspects.

### Skills to be acquired

- Data structures (linked lists, arrays)
- Loops and conditioning systems
- Job scheduler problems

### Teaching methods

The students work on one project for a total duration of 4 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

The project is to be done in a group of 2 to 4 students.

### Credit value

4 ECTS



## Assessment

Online project submission

## Project example

**Name:** Corewar

**Subject:** Create a parser that can change text-based champion into bytecode and a virtual machine (the arena) capable of interpreting the bytecodes to make the champions battle.

**Aim:** Parsing skills, understanding of virtual machines, scheduling

## [B-PSU-200] – Unix System Programming

**Unix System Programming** teaches on more advanced concepts linked to unix programming like signals and terminals.

### Skills to be acquired

- Signal
- Inter-process communication handling,
- Filesystem operations
- Interactive use of the terminal.

### Teaching methods

The students work on 2 projects for a total duration of 6 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

The projects are to be done in groups of up to 2 students.

### Credit value

4 ECTS

## Assessment

Online project submission

## Project example

**Name:** my\_sokoban

**Subject:** Re-create the sokoban game using terminal capabilities via the libncurses

**Aim:** Understand how a terminal works and how to use it's capabilities

## [B-PSU-210] – Shell Programming

**Shell Programming** focuses on more advanced aspects of programming of your own shell.

### Skills to be acquired

- Correctly program in C
- Understand and interact with the operating system
- Calmly understand a big project



- Manage a long-term group project

### Teaching methods

The students work on two projects for a total duration of 8 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

The minishell2 is a solo project, the 42sh in a group of 4 to 5 students.

### Credit value

5 ECTS

### Assessment

Online project submission

### Project example

**Name:** 42sh

**Subject:** Create your own shell capable of launching process, handling the environment, use builtins

**Aim:** Uses process, pipes, parsing.

## [B-MUL-200] – Graphical Programming

**Graphical programming** consists in creating images/animations and scenes management using algorithms and graphical resources.

This unit aims at teaching advanced skills linked to this topic and at synthetizing the experience into a final project: MyRPG

The two core axes "mathematics" and "game" are still existing, added to a new UX/UI-oriented aspect.

### Skills to be acquired

- Isometric/parallel projection
- Creation of UI elements (e.g. subwindows, buttons)
- Responsivity of user interactions (animations, color changes, layering)
- In-game balancing (skills, stuff)

### Teaching methods

The students work on two projects for a total duration of 11 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

The first project is to be done in pairs. The MyRPG project are in a group of 3 to 4 students.

### Credit value

9 ECTS

### Assessment

Online project submission

### Project example

**Name:** MyRPG

**Subject:** Create your own RPG game using the CSFML



## [B-AIA-200] – Introduction to Artificial Intelligence

**Introduction to Artificial Intelligence** is based on one single project, which consists of simulating an autonomous car.

It aims also at discovering the field of Artificial Intelligence: what is it about? what can we do with it?

### Skills to be acquired

- Overview of "Artificial Intelligence" field
- Introduction to research in a state-space
- Awareness of the data structure problems and algorithmic complexity
- Methodical approach to measuring a program's performance

### Teaching methods

The students work on one project for a total duration of 5 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

This project is to be done in pairs.

### Credit value

3 ECTS

### Assessment

Online project submission

### Project example

**Name:** need4stek

**Subject:** Move a simulated car in an autonomous way across various tracks

**Aim:** Basics of A.I.

## [B-WEB-200] – Introduction to Web Development

The **Introduction to Web Development** unit is composed of a single project that consists of creating a REST API that reads and writes to a database.

The goal is to introduce students to the field of web development

### Skills to be acquired

- the creation of a database
- understanding the VCM architecture
- the use of different tools (virtualenv / pip / Python3 / Flask / Jinja2 / MySQL)

### Teaching methods

The students work on one project for a total duration of 5 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

This project is to be in groups of 2 to 3 students.

### Credit value

2 ECTS



## Assessment

Online project submission

## Project example

**Name:** EpyTodo

**Subject:** Create a REST API to manage a ToDoList with CRUD operations.

**Aim:** Discovery of the basics of web development with python

## [B-DOP-200] – Introduction to DevOps

The **Introduction to DevOps** unit is composed of two projects whose goal is to introduce the basics of DevOps (an area at the crossroads of IT development and system administration) through Docker and an introduction to CI/CD through Github Actions.

### Skills to be acquired

- Build and deploy an existing project using docker and docker-composer
- Implement task automation on a git repository using Github Actions

### Teaching methods

The students work on two projects for a total duration of 6 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

### Credit value

2 ECTS

## Assessment

Online project submission

## Project example

**Name:** popeye

**Subject:** Deploy a project exist thanks to docker and docker-composer.

**Aim:** Discovery of the use and creation of docker configuration files.

## [B-MAT-200] - Mathematics

**Mathematics** focuses on the mathematical tools. The unit is a complementary introduction to programming (reviewing some of the Pool's elements) and is also an introduction to scientific programming.

### Skills to be acquired

- Linear algebra: vector analysis, matrix calculus
- Geometry: geometric transformations and coordinate systems
- Nonlinear equation solving

### Teaching methods

The students work on five mini-projects of two weeks each for a total duration of 10 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

The students work on each project alone or in pairs.



Credit value

2 ECTS

Assessment

Online project submission.

## {[B-MET-200] – Professional Competencies

**Professional competencies** is mainly composed of review sessions.

The reviews allow the students to present what they have completed for the project. They also must explain the reasons why they choose to follow this path and how they get to this result.

The code reviews allow the students to share programming experience and get feedbacks from their peers.

Skills to be acquired

Each review will be evaluated on 4 elements:

- Quality Assurance (unit tests, functional tests, no code defects, ...)
- Methodology and Organization (frequencies of commit, use of branches or external tools, group or personal organization, work methods, ...)
- Involvement (going that extra mile than what the project demands)
- Quality of oral presentation

Teaching methods

The reviews are open to any other students. An academic member evaluate it based on several criteria.

Code reviews are not evaluated. The evaluator helps the students on commenting each other code.

Credit value

3 ECTS including:

- 2 ECTS based on the participation and quality of the reviews.
- 1 ECTS based on the participation of at least two code reviews

Assessment

Oral presentation

# Epitech Second-Year Program

Only Spring semester available





from January to July

**Program aim:**

- Acquire an understanding of object-oriented programming via C++ language
- Expand your technical panorama by learning

**Learning outcome:**

- Deepen and broaden your knowledge and skills of tools to be autonomous and adaptable.

**Prerequisites:**

- Advanced knowledge of C language
- English language proficiency: TOEFL IBT: 65/IELTS:5.5/TOEIC:600 or English test of your own home institution equivalent to B2

**Courses list:**

<u>Code</u>	<u>Course</u>	<u>ECTS</u>	<u>Semester</u>
B-CPP-300	C++ Seminar	5	Spring
B-CNA-410	Computer Numerical Analysis – Trading	2	Spring
B-CCP-400	Concurrent Programming	2	Spring
B-FUN-400	Functional Programming	4	Spring
B-MAT-400	Mathematics	3	Spring
B-DOP-400	DevOps	2	Spring
B-NWP-400	Network Programming	4	Spring
B-OOP-400	Object-Oriented Programming	4	Spring
B-MET-400	Professional Competencies	3	Spring
B-PSU-402	Unix Programming - Instrumentation	3	Spring



<b>B-PSU-400</b> <b>Spring</b>	<b>Unix Programming - Memory</b>	<b>3</b>	
<b>B-ASM-400</b>	<b>x86-64 Assembly</b>	<b>2</b>	<b>Spring</b>
<b>B-YEP-400</b>	<b>Year-End-Project – Indie Studio</b>	<b>4</b>	<b>Spring</b>
<b>B-YEP-410</b>	<b>Year-End-Project - Zappy</b>	<b>4</b>	<b>Spring</b>
<b>B-SEC-400</b>	<b>Cyber Security</b>	<b>2</b>	<b>Spring</b>

## **Courses description:**

### **[B-CPP-300] C++ Seminar**

C++ Seminar (or Pool) is a key moment at Epitech. Like the C-Seminar module, it represents an intensive module and it is highly instructive.

The students learn about object-oriented programming through the fundamental elements of the C++ language. Classes, instances and methods are part of the module.

Skills to be acquired

- Correct usage of the C++ language and its special features in relation to C.
- Ability to produce a basic C++ program in Unix.
- Ability to be punctual and work regularly
- Ability to be precise in the reading, and comprehension, of one's code.

Teaching methods

All the materials are available online. Academic mentors help the students to go through. In addition to the knowledge and skills the students acquire additional skills in time management/team work as if they would be in a company. The module teaches the basic knowledge. Throughout, the students create hypotheses, runs tests while collaborating and sharing ideas between each other.

Credit value

5 ECTS

Assessments

Online project submission

Project Example

Name: Rush

Subject: Basic principles of C++ language



Aim: Train student to overcome the stress and pressure inherent to crunch time (important amount of work in a short period of time. Here, 2 days)

## [B-CNA-410] B4 – Computer Numerical Analysis – Trading

**Computer Numerical Analysis- Trading** introduces to the use of programming and mathematical tools in the field of trading.

### Skills to be acquired

- Research professional tools and how to implement them.
- Develop a custom algorithm
- Interface with an existing platform
- Algorithm optimization and live editing

### Teaching methods

The students work on two projects for a total duration of 18 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

### Credit value

2 ECTS

### Assessments

Online project submission

### Project Example

**Name: Trade**

**Subject:** Elaborate your own algorithm in a simulated trading environment

## [B-CCP-400] B4 – Concurrent Programming

**Concurrent Programming** teaches the concept of concurrent accesses in programming.

### Skills to be acquired

- Threads and mutexes
- IPC (semaphores, message queues, etc.)

### Teaching methods

The students work on one project for a total duration of 4 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project. The project is to be worked in groups of 2 to 4 students.

### Credit value

2 ECTS

### Assessments

Online project submission



## Project Example

**Name: The Piazza**

**Subject:** Create a program that handle a pizzeria capable of scaling by opening multiple kitchen and multiple cooks per kitchen.

## [B-FUN-400] B4 – Functional Programming

**Functional Programming** focuses on introducing the functional paradigm of programming.

### Skills to be acquired

- Recursive functions
- Lists
- Pattern matching and pattern guards
- Partial application
- Modules
- Project management with Stack
- Input/Output
- Monads

### Teaching methods

The students work on two projects for a total duration of 7 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

The second project is to be worked in pairs.

### Credit value

4 ECTS

### Assessments

Online project submission

## Project Example

**Name:**

**Subject:**

**Aim:**

## [B-MAT-400] B4 – Mathematics

**Mathematics** focuses on Probability and Statistics.

This unit is the continuation of the previous modules Mathematics. The students must take the required modules to attend the course. The students learn to create algorithms by using mathematical methods and to use graphic tools (plotting different kinds of curves).

### Skills to be acquired

- Probability



- Statistics

### Teaching methods

The students work on nine mini-projects of 2 weeks each for a total duration of 18 weeks of work on this unit. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

The students work on each project alone or by two.

### Credit value

3 ECTS

### Assessment

Online project submission.

### Project example

**Name:** 202unsold

**Subject:** Compute the features of random variables defined by their mutual probability law.

## [B-DOP-400] B4 – DevOps

The **DevOps** unit is composed of two projects whose goal is to go further into DevOps practices (an area at the crossroads of IT development and system administration).

### Skills to be acquired

- Creating and managing a set of Docker images and containers using Dockerfiles and Docker Compose
- Setting up a Jenkins instance for Continuous Integration and various automation tasks (such as automated testing) using Jenkins Configuration as Code and Jenkins Job DSL

### Teaching methods

The students work on two projects for a total duration of 6 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

### Credit value

2 ECTS

### Assessment

Online project submission

### Project example

**Name:** my\_marvin

**Subject:** Setup a jenkins instance using configuration-as-code.

## [B-NWP-400] B4 – Network Programming



**Network programming** is specifically designed to introduce the following concepts:

- Network communication
- Data packets

Skills to be acquired

- Multi-client programming
- Communication protocol implementing
- Documentation reading and understanding

Teaching methods

The students work on three projects for a total duration of 8 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team information about the completion of each project. The first two projects are solo project. The third project is in pairs.

Credit value

4 ECTS

Assessments

Online project submission

Project Example

**Name:** FTPServer

**Subject:** Create a server respecting the FTP protocol

**Aim:** Know how to read an RFC and introduction to TCP sockets.

## [B-OOP-400] B4 – Object-Oriented Programming

**Object-Oriented Programming** introduces to the Object-Oriented Programming paradigm.

It focuses on modularization and problematic' abstraction by using practical examples such as dynamic libraries.

Skills to be acquired

- Paradigm shift
- Modularization
- Abstraction and generic programming

Teaching methods

The students work on two projects for a total duration of 10 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team information about the completion of each project.

The first project is a group project for up to 2 students.

The second project is a group project of 2 to 3 students.

Credit value

4 ECTS

Assessments

Online project submission



## Project Example

**Name:** Arcade

**Subject:** Create an arcade system with a core capable of handling multiple game and multiple display library (terminal, 2D graphics, 3D graphics, ...)

**Aim:** Architecture a project around modules that can be interchanged.

## [B-MET-400] B4 – Professional Competencies

**Professional Competencies** is mainly based on review sessions.

The reviews allow the students to present what they have completed for the project. They also must explain the reasons why they choose to follow this path and how they get to this result.

The code reviews allow the students to share programming experience and get feedbacks from their peers.

### Skills to be acquired

Each review will be evaluated on 4 elements:

- Quality Assurance (unit tests, functional tests, no code defects, ...)
- Methodology and Organization (frequencies of commit, use of branches or external tools, group or personal organization, work methods, ...)
- Involvement (going that extra mile than what the project demands)
- Quality of oral presentation

### Teaching methods

The reviews are open to any other students. An academic member evaluate it based on several criteria. Code reviews are not evaluated. The evaluator helps the students on commenting each other code.

### Credit value

3 ECTS including:

- 2 ECTS based on the participation and quality of the reviews.
- 1 ECTS based on the participation of at least two code reviews

### Assessment

Oral presentation

## [B-PSU-402] B4 – Unix Programming - Instrumentation

**Unix Programming - Instrumentation** teaches the students to have a better understanding of ELF files and reverse engineering. It allows the students to understand how debuggers and disassemblers function in UNIX.

### Skills to be acquired

- Discern the userland's kernel space;
- Know which information can be retrieved in a process
- Explore the system call concept in greater depth
- Learn how to plot a program's execution
- Become an expert in Unix system programming
- Learn how to decode x86-64 binary instructions



### Teaching methods

The students work on two projects for a total duration of 6 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

The second project is in pairs.

### Credit value

3 ECTS

### Assessments

Online project submission

### Project Example

**Name:** strace

**Subject:** Re-code the strace program

**Aim:** master the basics of debugging in linux (ptrace)

## [B-PSU-400] B4 – Unix Programming - Memory

**Unix Programming- Memory** introduces about memory allocation management in an UNIX environment.

### Skills to be acquired

- Understand the mechanisms of memory management (in particular the links between physical memory and virtual memory);
- Understand the structure of a binary (ELF) file format

### Teaching methods

The students work on two projects for a total duration of 5 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project.

### Credit value

3 ECTS

### Assessments

Online project submission

### Project Example

**Name:** malloc

**Subject:** Create your own version of the function malloc and free

**Aim:** Learn about memory management using brk()

## [B-ASM-400] B4 – x86-64 Assembly

**x86-64 Assembly** introduces the x86-64 assembly and the use of this low-level language for the development of a minimalistic C library.

### Skills to be acquired

- Know about x86-64 processor and its instruction set
- Know about memory and stack operation
- Know about address spaces and function calling

### Teaching methods

The students work on one project for a total duration of 4 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each





project.

The project is to be worked in pairs.

Credit value

2 ECTS

Assessments

Online project submission

Project Example

**Name:** minilibc

**Subject:** Create your own miniature version of the LibC

### [B-YEP-400] B4 – Year-End-Project – Indie Studio

**Year-End-Project – Indie Studio** corresponds to one of the two final projects of the semester. The first project is linked to the understanding of an OOP architectures.

Skills to be acquired

- Go from procedural paradigm to object paradigm
- Modularize a problematic
- Abstract such problematic

Teaching methods

The students work on one big project for a total duration of 6 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team information about the completion of each project. The project is to be done in groups of 4 to 6 students.

Credit value

4 ECTS

Assessments

Online project submission

Project Example

**Name:** Indie Studio

**Subject:** As an independent game studio would, re-create a classic Bomberman game.

### [B-YEP-410] B4 – Year-End-Project - Zappy

**Year-End-Project – Zappy** is one of the two final projects of the semester. It summarizes several concepts such as network programming (using TCP sockets), Artificial Intelligence and GUI.

Skills to be acquired

- Network programming
- GUI
- Basic artificial intelligence



### Teaching methods

The students work on one project for a total duration of 6 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team the information about the completion of each project. The project is to be done in groups of 4 to 6 students.

### Credit value

4 ECTS

### Assessments

Online project submission

### Project Example

**Name: Zappy**

**Subject:** Simulate a world (the TCP server) and it's habitants in a quest to survive and evolve.

## [B-SEC-400] B4 – Cyber Security

The **Cyber Security** unit introduces students to new notions regarding Cyber Security. It is a follow-up to the first year unit.

### Skills to be acquired

- Usage of steganography tools
- Understanding and exploiting of binaries (buffer overflow, ...)

### Teaching methods

A "Capture The Flag" tournament in which students are asked to take on security challenges, demonstrating their skills.

### Valeur en crédit(s)

2 ECTS

### Assessments

The evaluation is based on the number of points and challenges validated on the CTF platform and a theoretical evaluation.

# Epitech third-year program

## Fall semester/Academic year



From September 2022 to July 2023

### **Program aim:**

- Acquire more advance computer skills
- Work and manage a middle size group
- Introduction to software architecture and design pattern

### **Learning outcomes:**

Students will be able to tackle more advanced algorithmic problems and build middle to large size software.

### **Prerequisites:**

- C programming language
- C++ programming language
- Object Oriented Programming
- Design pattern (basic knowledge)
- English language proficiency equivalent to B2 (TOEFL IBT: 65 / IELTS: 5.5/ TOEIC: 600, or English test of their institution equivalent to B2

### **Courses list:**

<b>Code</b>	<b>Course</b>	<b>Credit (ECTS)</b>	<b>Semester</b>
B-SEC-500	Advanced Cyper Security	2	Fall
B-DOP-500	DevOps	3	Fall
B-MAT-500	Mathematics	3	Fall
B-FUN-510	Functional Prog – evalExpr	2	Fall
B-FUN-501	Functional Prog – HAL	2	Fall
B-AIA-500	Artificial Intelligence	2	Fall
B-DEV-500	AppDev – Dashboard	2	Fall
B-DEV-501	AppDev – Epicture	2	Fall
B-CPP-501	Advanced C++ - R-Type	3	Fall



<b>B-CPP-500</b>	<b>Advanced C++ - Babel</b>	<b>2</b>	<b>Fall</b>
<b>B-MOO-500</b>	<b>Innovation – Moonshot</b>	<b>3</b>	<b>Fall</b>
<b>B-INN-000</b>	<b>Guided Project – Innovation Hub</b>	<b>6</b>	<b>Fall/Spring</b>
<b>M-FLE-000</b>	<b>French Language course I</b>	<b>2</b>	<b>Fall</b>
<b>B-YEP-510</b>	<b>Advanced C++ - ZIA</b>	<b>5</b>	<b>Spring</b>
<b>B-YEP-510</b>	<b>App-Dev – AREA</b>	<b>5</b>	<b>Spring</b>
<b>B-YEP-510</b>	<b>Functional Prog – KOAK</b>	<b>6</b>	<b>Spring</b>
<b>M-FLE-000</b>	<b>French Language course II</b>	<b>2</b>	<b>Spring</b>

### **Course description :**

#### **[B-SEC-500] – Advanced Cyber Security**

**Advanced Cyber Security** is the follow up to the second year unit. It goes deeper and introduced advanced notions.

#### Skills to be acquired

- Exploiting Windows and Linux (Kerberos, samba, nfs, ...)
- Pivoting
- Network sniffing (wireshark, ssh tunnelling)

#### Teaching methods

A "Capture The Flag" tournament in which students are asked to take on security challenges, demonstrating their skills.

#### Credit value

2 ECTS

#### Assessment

The evaluation is based on the number of points and challenges validated on the CTF platform and a theoretical evaluation.

#### **[B-DOP-500] – Advanced DevOps**

**Advanced DevOps** teach the students the deeper concepts of DevOps.

#### Skills to be acquired

- Build and deploy a complete web app using docker and docker-compose
- Deploy and configure a Jenkins instance with CI/CD
- Scale a service over a cluster using docker swarm and treafik



### Teaching methods

The students work on 3 mini-projects for a total duration of 6 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team information about the completion of each project. The mini projects are to be done alone; the project is to be done in groups.

### Credit value

3 ECTS

### Assessment

Project submission

### Project example

**Name:** popeye

**Subject:** Deploy a project exist thanks to docker and docker-composer.

**Aim:** Discovery of the use and creation of docker configuration files.

## [B-MAT-500] - Mathematics

**Mathematics** studies the advanced scientific calculation algorithms in operations research. Operation research represents all the methods and models that allow numerous business management and organization issues to be clarified and solved. The objective of the module is to introduce notions of the most important algorithms in scientific calculations which are used in operations research.

### Skills to be acquired

- Complexity of algorithms
- Graph theory
- Linear systems
- Program and interpolation

### Teaching methods

During the module, the students work on 9 mini-projects. The projects cover three topics:

- Algorithm through projects n°301, n°305, n°307
- Data structures (matrix and graphs) through projects n°302, n°303, n°304, n°306
- Interpolation through projects n°308 and n°309

The module work as an inter-disciplinary module. The students are evaluated also on their professional behavior, detailed and effective work and their involvement.

### Credit value

3 ECTS

### Assessment

Project submission

## [B-FUN-500] - Functional Prog – evalExpr

**Functional Prog – evalExpr** is an introduction to advanced functional concepts. The students work on abstract concepts such as monads or lambdas. It is an introductory module for the module B-FUN-501. 11



#### Skills to be acquired

- Advanced functional concepts

#### Teaching methods

The students work on 1 project for a total duration of 2 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team information about the completion of each project. The project is to be done in groups of 2 students.

#### Credit value

2 ECTS

#### Assessment

Project submission

### [B-FUN-501] - Functional Prog – HAL

**Functional prog – HAL** is an introduction to advanced functional concepts. The students work on abstract concepts such as monads or lambdas.

#### Skills to be acquired

- Advanced functional concepts

#### Teaching methods

The students work on 1 project for a total duration of 7 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team information about the completion of each project. The project is to be done in groups of 2 students.

#### Credit value

2 ECTS

#### Assessment

Project submission

### [B-AIA-500] - Artificial Intelligence

**Artificial intelligence** is based on the Gomoku project. It focuses on the decision-making process in a two player game. The notions of Minimax theory and Monte Carlo methods will be discussed as well as Machine Learning through genetic algorithms and artificial neural network.

#### Skills to be acquired

- Knowledge representation (how to define and complete an efficient goal-driven data structure
- Minimax and Monte Carlo methods or equivalent methods (understand the methods usage in a decision process within a two-player strategy and to be able to choose one of them)
- Heuristic thinking and implement a non-static heuristic that efficiently estimate the solutions
- Run test on programs which aim at the best efficiency

#### Teaching methods

Through the Gomoku project, the students must:

- Formalize the subject matter
- Define efficient structures



- Implement a decisional algorithm
- Create a complete Gomoku AI which is able to deploy and to adapt strategies
- Comply with an existing process

Credit value

2 ECTS

Assessment

Project submission

Project example

**Name:** Gomoku

**Subject:** game theory algorithm and basic AI

**Aim:** Learn and implement basic game theory algorithms like min-max through an easy but challenging game. The game must be developed from scratch

## [B-DEV-500] - AppDev – Dashboard

**AppDev – Dashboard** focuses on the most used programming languages and the most used ecosystems in the today's industry.

Skills to be acquired

- Able to understand the concepts of the chosen language
- Able to use build tools and dependency managers used by most companies
- Able to understand how to use and to create a web service

*Programming languages*

- Java
- C#
- .Net
- Javascript (via NodeJS)

*Tools*

- Maven
- JUnit
- NPM
- NuGet

*Major Libraries*

- Netty
- Protocol Buffers

Teaching methods

Through the module, the students learn the programming languages and tools through the creation and the use of the web service.

Credit value

2 ECTS

Assessment

Project submission



## [B-DEV-501] - AppDev – Epicture

**AppDev – Epicture** focuses on the most used programming languages and the most used ecosystems in the today's industry.

### Skills to be acquired

- Knowledge of the Android/UWP development environment
- Knowledge of the tools and processes to develop and test projects
- Able to understand and to use APIs through web services
- Able to evaluate and understand the user experience and the user interface through your program developments

### Programming languages

- Java
- C#
- .Net
- Javascript (via NodeJS)

### Tools

- Maven
- JUnit
- NPM
- NuGet

### Major Libraries

- Netty
- Protocol Buffers

### Teaching methods

In the module, the students learn how to create a client application (either mobile application or desktop application) via UWP or Android. It is considered the students to have the required basics of the chosen development stack and to use the specific aspects related to the Android/UWP development. The project consists of creating a client application for a well-known pictures service using its API.

### Credit value

2 ECTS

### Assessment

Project submission

## [B-CPP-501] - Advanced C++ - R-Type

**Advanced C++- R-Type** teaches the deep aspects of the architecture of a C++ program. It consists of introducing the philosophic difference between UNIX and Windows in order to create software abstractions which allow originally portable programs to be developed in all systems. The abstraction focuses on a key concept leading the whole topics: Application Programming Interfaces (API) are elements which are automatically linked to Object Oriented Programming, all programming languages combined.

### Skills to be acquired

- Able to understand the differences between Unix and Windows system
- Able to identify the mistakes/attributes of these operating systems
- Able to find and read Windows information: MSDN
- Able to know how to use and to configure Visual Studio, Microsoft's IDE
- Able to know how to use Visual Studio's extraordinary debugger





- Able to know how to create a whole UML class diagram (used by all object languages)
- Able to understand, to use and to design an API
- Able to use an abstract shared library in Windows and UNIX
- Able to use abstract sockets in Windows and UNIX
- Able to use abstract threads in Windows and UNIX

### Teaching methods

The students must make a copy of the R-Type game. The projects are based on an industry use of C++. The purpose of the module is the projects to be built and to run it on Unix as well as on Windows systems.

### Credit value

3 ECTS

### Assessment

Project submission

### Project example

**Name:** R-Type

**Subject:** Game theory and Architecture as for Scripting

**Aim:** Develop a well-known retro game called R-Type as an initiation to game development, program architecture and scripting API

## [B-CPP-500] - Advanced C++ - Babel

**Advanced C++ - Babel** teaches the deep aspects of the architecture of a C++ program. It consists of introducing the philosophic difference between UNIX and Windows in order to create software abstractions which allow originally portable programs to be developed in all systems. The abstraction focuses on a key concept leading the whole topics: Application Programming Interfaces (API) are elements which are automatically linked to Object Oriented Programming, all programming languages combined.

### Skills to be acquired

- Able to understand the differences between Unix and Windows system
- Able to identify the mistakes/attributes of these operating systems
- Able to find and read Windows information: MSDN
- Able to know how to use and to configure Visual Studio, Microsoft's IDE
- Able to know how to use Visual Studio's extraordinary debugger
- Able to know how to create a whole UML class diagram (used by all object languages)
- Able to understand, to use and to design an API
- Able to use an abstract shared library in Windows and UNIX
- Able to use abstract sockets in Windows and UNIX
- Able to use abstract threads in Windows and UNIX

### Teaching methods

The students must make a VOIP client/server program such as Skype. The projects are based on an industry use of C++. The purpose of the module is the projects to be built and to run it on Unix as well as on Windows systems.

### Credit value

2 ECTS



## Assessment

Project submission

## Project example

**Name:** Babel

**Subject:** tiny VoIP software

**Aim:** Develop VoIP application like Skype through

## [B-MOO-500] – Innovation - Moonshot

The objective of the **Moonshot** is to teach the students how to think differently, how to comprehend and to evaluate the topics and the societal/economic problems as opposed to strictly looking at the technical problems.

### Skills to be acquired

- Identify problems, and “difficult” issues within societal topics.
- Demonstrate the ability to apply solutions to problems encountered.
- Make the connection between a current societal topic and the solutions that I.T. can provide.
- Work consistency

### Teaching methods

The first two weeks focus on the understand of problem solving via the conferences that begin on the very first day. The conferences will be led by external speakers, who are well-known in their fields of expertise.

### Credit value

3 ECTS

## Assessment

Oral presentation

## [B-INN-000] Guided Project- Innovation Hub

Guided project teaches the students to practice all their knowledge and skills acquired in a business environment. They will be advised and supervised by the Innovation Center of Epitech.

The module starts with an introductory session with all the international students.

### Skills to be acquired

- Project management
- Ideation and brainstorming
- Prototyping
- Documentation
- Communication and persuasion skills

### Teaching methods

Guided project with monthly follow-ups supervised by the Epitech Innovation Center Team.



In parallel, the students must attend several activities of his choice in topics of IT offered by the Innovation Center to validate the module.

Credit value

6 ECTS

Assessment

Project submission

### [B-YEP-500] – Year End Project – Advanced C++ - ZIA

**Advanced C++ - ZIA** teaches the deep aspects of the architecture of a C++ program. It consists of introducing the philosophic difference between UNIX and Windows in order to create software abstractions which allow originally portable programs to be developed in all systems. The abstraction focuses on a key concept leading the whole topics: Application Programming Interfaces (API) are elements which are automatically linked to Object Oriented Programming, all programming languages combined.

Skills to be acquired

- Able to understand the differences between Unix and Windows system
- Able to identify the mistakes/attributes of these operating systems
- Able to find and read Windows information: MSDN
- Able to know how to use and to configure Visual Studio, Microsoft's IDE
- Able to know how to use Visual Studio's extraordinary debugger
- Able to know how to create a whole UML class diagram (used by all object languages)
- Able to understand, to use and to design an API
- Able to use an abstract shared library in Windows and UNIX
- Able to use abstract sockets in Windows and UNIX
- Able to use abstract threads in Windows and UNIX

Teaching methods

The students must make a HTTP server program such as a lightweight Apache. The projects are based on an industry use of C++. The purpose of the module is the projects to be built and to run it on Unix as well as on Windows systems.

Credit value

5 ECTS

Assessment

Project submission

### [B-YEP-500] – Year-End Project - AppDev – AREA

**AppDev – AREA** focuses on the most used programming languages and the most used ecosystems in the today's industry.

Skills to be acquired

- Able to understand the concepts of the chosen language
- Able to use build tools and dependency managers used by most companies
- Able to understand how to use and to create a web service
- Able to evaluate and understand the user experience and the user interface through your program developments



#### *Programming languages*

- Java
- C#
- .Net
- Javascript (via NodeJS)

#### *Tools*

- Maven
- JUnit
- NPM
- NuGet

#### Teaching methods

In the module, the students learn how to create a full service capable of connecting multiple existing APIs (a IFTTT-like) both front and back-end.

#### Credit value

5 ECTS

#### Assessment

Project submission

### [B-YEP-500] – Year-End Project - Functional Prog – KOAK

**Functional prog – KOAK** is the last unit of this introduction to advanced functional concepts. The students work on abstract concepts such as monads or lambdas.

#### Skills to be acquired

- Parsing using the functional paradigm
- Usage of AST
- Inferring types
- Code compilation

#### Teaching methods

The students work on 1 project for a total duration of 8 weeks of work. Each project is evaluated individually with automated tests giving the student and the academic team information about the completion of each project. The project is to be done in groups of 3 to 4 students.

#### Credit value

6 ECTS

#### Assessment

Project submission

### [M-FLE-000] French I Language Course

The **French** module teaches the students French (writing, listening, speaking) from the beginning level

#### Credit value

2 ECTS



Assessments

Class attendance and final exam