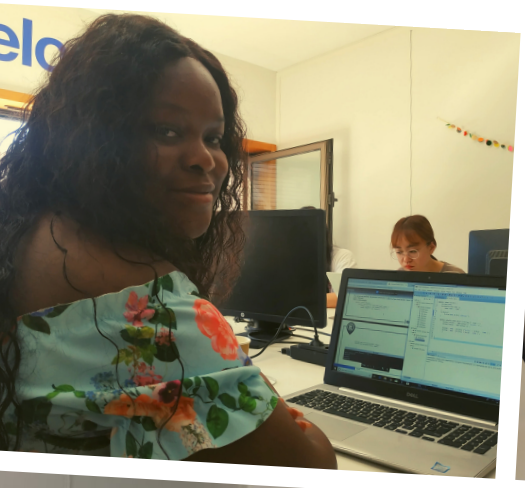
EPITECH.

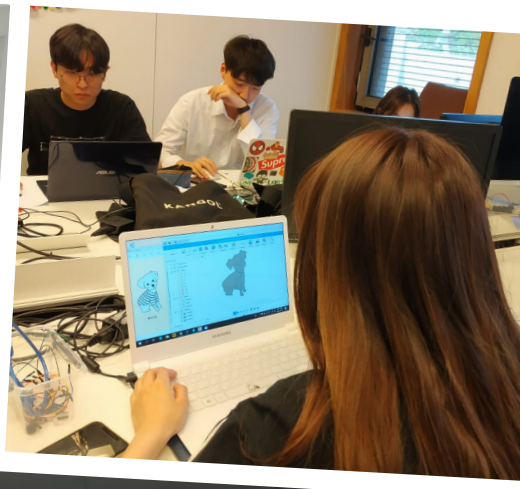
JOIN US AT

EPITECH - SCHOOL OF IT & INNOVATION

Courses offered in English (2020-2021)

FULL ACADEMIC YEAR



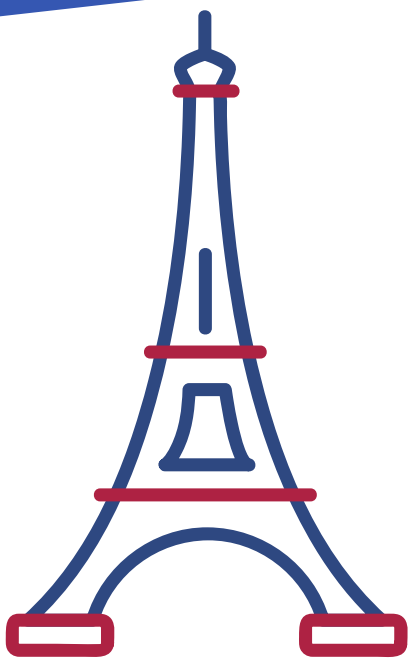


EPITECH

School of IT and Innovation in France

- Epitech as the largest French IT school
- Based on 13 campuses in France
- Also located on 4 european campuses (Spain, Germany, Belgium and Albania)

- Full project-based learning method implemented
- Bachelor level courses fully in English
- French language courses offered



Contact us on international-relations@epitech.eu

FULL ACADEMIC YEAR PROGRAMS

September 2020 to July 2021

Epitech International Bachelor Program

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:

Code	Course	Credit (ECTS)
B-CPE-100	Unix & C Lab Seminar I	4
B-CPE-101	Unix & C Lab Seminar II	3
B-CPE-110	Elementary Programming in C	5
B-PSU-100	Unix System Programming	5
B-MUL-100	C Graphical Programming	5
B-MAT-100	Mathematics	2
B-INN-000	Guided Project – Innovation Hub	6
M-FLE-000	French Language course	2
G-SEC-000	Binary Security	2
B-CPE- 200/201	Elementary Programming in C	9

B-PSU-200	**Unix System Programming**	**4**
B-PSU-210	**Shell Programming**	**5**
B-MUL-200	**Graphical Programming**	**9**
B-AIA-200	**Introduction to AI**	**3**
B-NSA-200	**Network and System Adm**	**2**

Courses description

[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 you to implement the initial concepts acquired, and especially will enable you to state your 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-.

Skills to be acquired

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

Teaching methods

All of the materials are available online. Academic mentors help you to go through the module. **Unix & C Lab Seminar I** represents 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 student creates hypotheses, runs tests while collaborating and sharing ideas with his classmates.

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. Here, 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 you to implement the initial concepts acquired and put in practice in the exercise of Bistromatic.

Skills to be acquired

- C Language

Teaching methods

Following the first part of the module, the student works on a stumper – project done by student pairs 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 start their curriculum.

Credit value

3 ECTS

Assessment

Project submission

Project example

> **Name:** Bistromatic
> **Subject:** Basic principles of C language
> **Outcome: Infinite Calculator**
> **Aim:** Tests students C language skills acquired during the Pool in a small but also complex project. Students will learn to look and implement well known algorithm to solve the given problem

[B-CPE-110] Elementary Programming in C

Elementary Programming in C is composed of two mini-projects Pushswap (individual project) and Lem-in (in group project).

Skills to be acquired

- Ability to solve complex technical and algorithmic problems
- Ability to analyze and solve a subject matter efficiently
- Ability to solve approximately 75% of the exercises in the allotted time period
- Ability to be precise and consistent on a subject matter

Teaching methods

The students work on two projects individually and in group.

Credit value

6 ECTS

Assessment

Project submission

Project example

> **Name:** Lem-In
> **Subject:** Basic shortest past
> **Outcome:** Shortest path algorithm implementation
> **Aim:** Understand and implement a path finding algorithm using newly mastered data structure (linked list)

[B-PSU-100] Unix System Programming

Unix System Programming covered all the fundamentals elements of UNIX programming and the development of the building blocks of a shell.

Skills to be acquired

- Ability to execute system calls
- Ability to solve algorithmic problems
- Ability to manage a project
- Ability to understand FileSystem
- Ability to execute binary files
- Ability for process management

Teaching methods

The module is divided in two parts. The first covers the fundamental elements of UNIX programming. The second part, the student will learn on how to develop the building blocks of your own shell.

For the first part, the students work on file and repository management, as well as on the advanced management of terminals and argument lists.

Credit value

6 ECTS

Assessment

Project submission

[B-MUL-100] C Graphical Programming

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

The objective of the module is to introduce you to graphical programming.

Skills to be acquired

For gaming projects

- Ressources, event and windows management
- Persistent data storing (e.g highscore, progression)
- Simple starting and pause menus

- Implementation of gameplay mechanics
- Implementation of simple game physics mechanics
- Famous visual effects (e.g parallax)

For mathematics projetcs

- 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 two important axis 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

Project submission

Project example

A list of projects will be available according to the topics selected by the student (mathematics or gaming). Example of projects:

Name: MyScreensaver
Subject: Animation based on light ot any visual effects and using particles
Aim: Introduction to C Graphical Programming

[B-MAT-100] Mathematics

Mathematics focuses on mathematics tools. This module is a complementary introduction to programming and an introduction to scientific programming.

Skills to be acquired

- Linear algebra: vector analysis, matrix calculus
- Geometry: geometric transformations and coordinate systems.
- Nonlinear equation solving
- Ability to master simple algorithms

Teaching methods

The students will work on projects which includes a number of skills to be acquired. It gives a solid basics of vector calculus and matric calculus.

Credit value

2 ECTS

Assessment

Project submission

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

Guided Project teaches the students to practice all the knowledge and the skills acquired in a business environment.

They will be advised and supervised by the Innovation Center of EPITECH.

Skills to be acquired

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

Teaching methods

Guided project with monthly follow-up supervised by the EPITECH Innovation Center team.

Credit value

5 ECTS

Assessment

Project submission

[G-SEC-000] B0 – Binary Security

Binary Security focuses on the vulnerabilities associated with application development.

Skills to be acquired

- Manipulate the reverse/debug tools: R2, GDB, Medusa etc.
- Knnow the basics of taking control of a program's execution flux.
- Understand how a simple algorithm works by reading its assembler code.
- Overrule the anti-debug techniques.
- Overrule a system's protections.

Teaching methods

A Call For Papers (CFP) starts the unit; it is about binary security or computer security in general. It is followed by a Capture The Flag tournament in which the students must complete security challenges.

Credit value

2 ECTS

Assessment

It is based on the number of points and challenges completed on the CTF platform.

[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 7 weeks. Each project is evaluated individually with automated tests. It gives the student and the academic team the information about the completion of each project.
A project is to be made in groups 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: lemin
Subject: Find the best way to move ants across the anthill
Aim: Pathfinding in a graph

[B-CPE-201] B2 – 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 4 weeks. Each project is evaluated individually with automated tests. It gives 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] B2 – Unix System Programming

Unix System Programming course teaches 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 two projects for 6 weeks. Each project is evaluated individually with automated tests. It gives 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: navy
Subject: Re-create the famous Battleship games using signals to communicate between processes
Aim: Inter-process communication using signals

[B-PSU-210] B2 – Shell Programming

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

Skills to be acquired

- C Programming
- Understand and interact with the operating system
- Manage a long-term group project
- Problem-solving approach for large projects

Teaching methods

The students work on two projects for 8 weeks. Each project is evaluated individually with automated tests. It gives the student and the academic team the information about the completion of each project.
The minishell2 is a solo project, the 42sh project is in groups 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] B2 – Graphical Programming

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

This unit aims at teaching advanced skills linked to graphical programming and at synthetizing the experience into a final project: MyRPG.

"Mathematics" and "game" are still the two core axes, added to a new UX/UI-oriented aspect in this new module on graphical programming.

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 11 weeks. Each project is evaluated individually with automated tests. It gives 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] B2 – 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 the "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 5 weeks. Each project is evaluated individually with automated tests. It gives 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-NSA-200] B2 – Network and System Administration
Network and System Administration teaches the students to master the 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 for 4 weeks. Each project is evaluated individually with automated tests. It gives 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_web
Subject: Install a small environment of VM with various services
Aim: Know how to make a dual boot installation and basic network configurations

FULL ACADEMIC YEAR PROGRAMS

September 2020 to January 2021

Epitech International Program

Third-year program

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:

Code	Course	Credit (ECTS)
B-SEC-500	**Security – Cryptography**	**2**
B-NSA-500	**DevOps**	**3**
B-MAT-500	**Mathematics**	**3**
B-FUN-510	**Functional Prog – evalExpr**	**1**
B-FUN-501	**Functional Prog – HAL**	**3**
B-AIA-500	**Artificial Intelligence**	**2**
B-DEV-500	**AppDev – Dashboard**	**2**

B-DEV-501	AppDev – Epicture	2
B-CPP-501	Advanced C++ - R-Type	3
B-CPP-500	Advanced C++ - Babel	2
B-INN-600	Guided Project – Innovation Hub	8
B-CPP-510	Advanced C++ - ZIA	5
B-DEV-510	App-Dev – AREA	5
B-FUN-510	Functional Prog – KOAK	6
M-FLE-000	French Language course	2

Courses description:

[B-SEC-500] Security – Cryptography

Security – Cryptography is a module on the ciphers and the encryptions: secure algorithm revealed. The CAESAR project is based on the crypto challenges which the level of difficulty increases step by step. The students start to work on the basics of Xoring and finish the module by breaking real-life cryptography.

The students do not need prior advanced knowledge in mathematics or in cryptography.

Skills to be acquired

- Ciphers
- Encryptions
- Break simple XOR cipher (CAESAR)
- Break repeating key XOR (vigenere)
- Attacks on AES in ECB and CBC mode

Teaching methods

The students start to work on basic project in cryptography and complete the module by getting strong knowledge in cryptography. It requires strong knowledge in programming. The compulsory programming languages to use are python language (strongly advised) or C or C++ language programming (more difficult).

Credit value

2 ECTS

Assessment

Project submission

[B-DOP-500] DevOps

DevOps teaches the students the basics of this practices by setting up docker containers and by doing an automated deployment in a scalable way.

Skills to be acquired

- Automate the provision of a server/VM using Ansible
- Build and deploy a complete web app using docker and docker-compose
- Scale a service over a cluster using docker swarm and treafik

Teaching methods

The students work on 2 mini-projects and 1 project for a total duration of 9 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 of 2 to 3 students.

Credit value

3 ECTS

Assessment

Project submission

Project example

Name: docker
Subject: Create a docker configuration to create a container capable of running a pre-made application
Aim: Discovres Dockerfile and docker-compose configuration and the tips and tricks to create containers.

[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-510] 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.

Skills to be acquired

- Advanced functional concepts

Credit value

1 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 a seven-week project. The project is evaluated via automated tests. The automated tests will give the academic team the information on the completion of the project. The project is to be done by pairs of students.

Credit value

3 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 about 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++- 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 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-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 will start with Moonshot. Moonshot is a two-week introduction

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.

Credit value

9 ECTS

Assessment

Project submission

[B-CPP-510] B5 - 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

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

Credit value

5 ECTS

Assessment

Project submission

[B-DEV-510] B5 - 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

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-FUN-510] B5 - 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 a project for 8 weeks. Each project is evaluated individually with automated tests. It gives 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