

Préambule à CSC4526/JIN4 (Développement C++)

Michel Simatic

Télécom SudParis

30 avril 2025

Plan

1. [Introduction](#)
2. [CSC4526/JIN4 : une UV inscrite dans la VAP JIN](#)
3. [Zoom sur CSC4526/JIN4](#)
4. [Conclusion](#)

Introduction

1. → [Introduction](#)
2. [CSC4526/JIN4 : une UV inscrite dans la VAP JIN](#)
3. [Zoom sur CSC4526/JIN4](#)
4. [Conclusion](#)

Qui est qui ?

1. Sur Wooclap <https://app.wooclap.com/CPPTSP>, en 3 fois **un** mot (séparés par des Retours à la ligne) :
 1. Que vous évoque C++ ?
 2. Votre attente par rapport à CSC4526/JIN4 (pas JIN)
 3. Votre crainte par rapport à CSC4526/JIN4 (pas JIN)
2. Debout, depuis votre place :
 1. Prénom NOM
 2. Vos 3 mots (envoyés à Wooclap quand vous vous levez)
 3. Où vous voyez-vous dans 5 ans ?
 4. Quels langages maîtrisez-vous bien ?
 5. Freins à CSC4526/JIN4 (par exemple, Cassiopée dont l'avancement est insuffisant, CF2, etc.)
3. Après cette séance, sondage Moodle sur votre parcours de 2A avant d'arriver en CSC4526 et votre Cassiopée

CSC4526/JIN4 : une UV inscrite dans la VAP JIN

1. [Introduction](#)
2. → [CSC4526/JIN4 : une UV inscrite dans la VAP JIN](#)
3. [Zoom sur CSC4526/JIN4](#)
4. [Conclusion](#)

Organisation de la VAP JIN

- S8 spécifique TSP
 - CSC4508 : Systèmes d'exploitation
 - → CSC4526 : Développement C++
- S9 (TSP & ENSIIE)
 - Dans les locaux de l'ENSIIE et de TSP
 - Ensemble d'UE à la structure ~ filière ENSIIE classique
 - IMA5061-MORE : Modélisation Et Rendu (1^{er} 1/2-semester)
 - CSC5063-DMIA : (Seulement ENSIIE ? Dév sur Mobile) & IA (2^e 1/2-semester)
 - CSC5062-INCA : Interactions Et Capteurs (1^{er} 1/2-semester)
 - CSC5064-SYIR : Systèmes Interagissant en Réseau (2^e 1/2-semester)
 - CSC5065-PFE: Projet d'approfondissement (semester)
 - DIV5010-PROCCI : Profils de Comportement et de Comm. (17-21/11 et avant)
 - DIV5061-SHS : SHS, Design & Game Design (1^{er} 1/2-semester et après)
 - ENV5001 (TSP) : Initiation à la Resp. Sociétale des Entreprises (semester)
- S10
 - Stage 6 mois en entreprise ou laboratoire (à partir du TODO 10/02/25)
 - Géré par l'école d'origine de l'étudiant·e (quid cas étudiant·e·s en mobilité ?)

Planification S9 JIN : Dates phare

- Proposition de Projet de Fin d'Etude = été 2025
- Devoir de vacances = été 2025
 - On suppose Unity OK
 - Étudiant·es TSP : Rattrapage par rapport à VR
 - Étudiant·es ENSIIE : Rattrapage par rapport à C++
 - Si possible, vous auto-former à Unreal Engine
 - Révisions sur les produits scalaires (pour cours de rendu)
 - (Optionnel) lié à PROCCI, lecture de 1-2 livres
- Rentrée S9 = 08/09/25 ? à 9h (TSP)
 - 08/09 : Après-midi « Intégration »
- Congés d'automne = 27/10/25 – 02/11/25
- Paris Games Week : JIN n'y participe plus
- PROCCI = 17/11/25 – 21/11/25
- UniJam = ??/??/25 18h – ??/??/25 20h
- Congés Noël = 22/12/25 – 04/01/26
- Fin des cours JIN = 05/02/26 ? à 13h

Zoom sur CSC4526/JIN4

1. [Introduction](#)
2. [CSC4526/JIN4 : une UV inscrite dans la VAP JIN](#)
3. → [Zoom sur CSC4526/JIN4](#)
4. [Conclusion](#)

Acquits d'apprentissage visés

- A l'issue du module, les étudiants sont capables, dans le cadre d'une application interactive (qui peut être un jeu vidéo) de petite taille, mais réaliste de :
 - spécifier cette application sur la base d'un sujet imposé
 - développer cette application en C++ moderne (utilisation de "Resource Acquisition Is Initialization"-RAII, de pointeurs intelligents), à l'aide de l'environnement de développement (IDE) Visual Studio ou CLion
 - développer des tests unitaires dans le cadre d'une démarche Test-Driven Development (TDD)
 - intégrer 3 bibliothèques externes (dont le moteur graphique SFML ou Qt) à leur application,
 - mettre en œuvre une hiérarchie de classes et expliquer cette hiérarchie de classe à l'aide d'un diagramme UML,
 - intégrer au moins un design pattern (qui peut être un game programming pattern) au sein de l'application,
 - implanter une interface utilisateur intuitive et fluide,
 - présenter et/ou critiquer de manière argumentée, en public, un travail réalisé.

Équipe pédagogique

- Loïc Joly (SonarSource - Expert C++)



- Philipp Schlehuber-Caissier (TSP - Intervenant féru de C++)



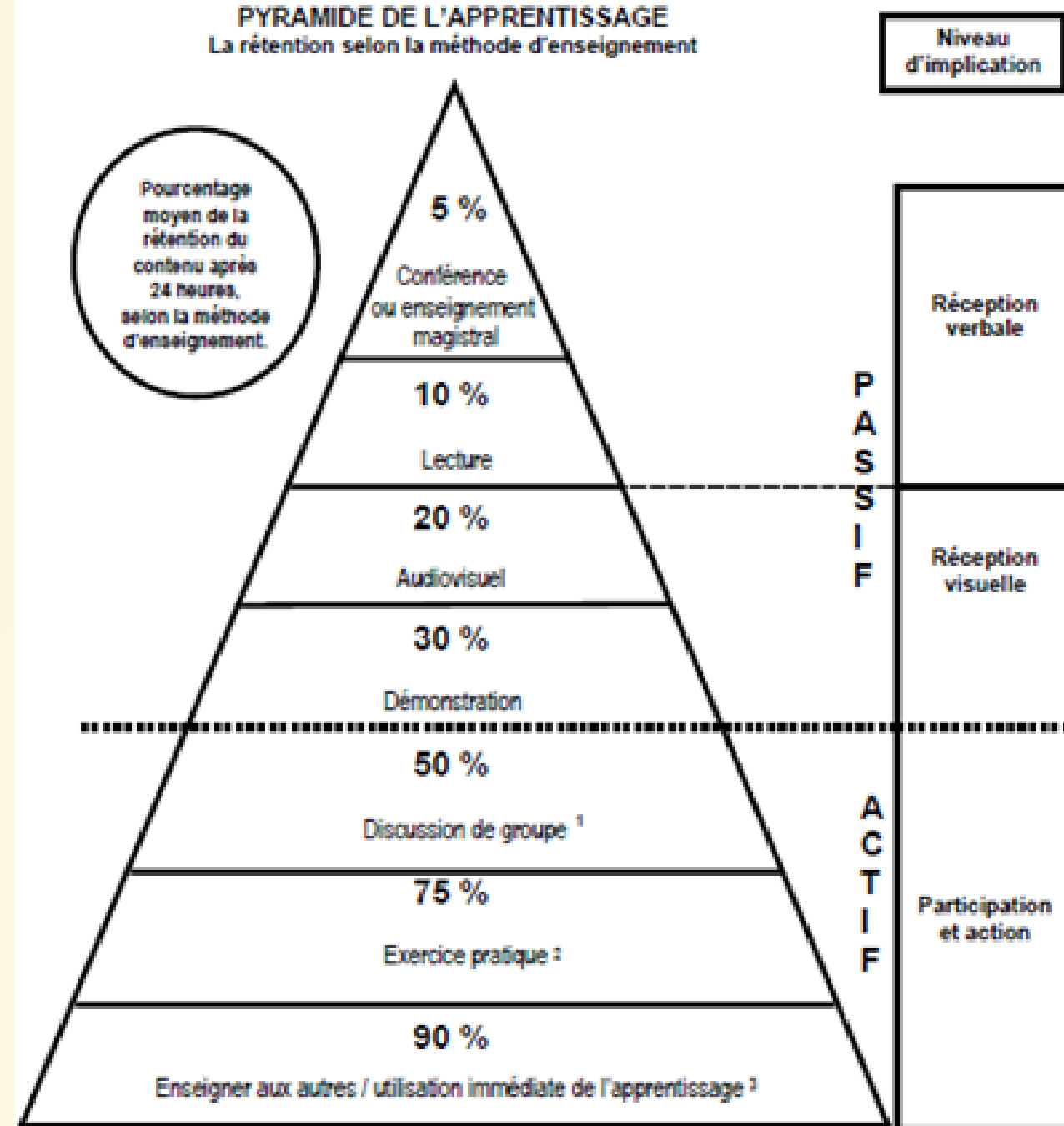
- Michel Simatic (TSP - Coordonnateur)



Grandes lignes

- Modalités
 - Un peu de théorie
 - Beaucoup de pratique (notamment avec un exercice fil rouge)
 - Un micro-projet (jeu ou application interactive, au choix)
- Pré-requis
 - Algorithmique (notions)
 - Langage C (bonne pratique)
 - Java / Lang. objet (bonne pratique)
 - UML (bonne pratique)

Préambule à CSC4526/JIN4 (Développement C++)



Source : National Training Laboratories of Bethel (Maine, USA) à partir de résultats de recherches sur la mémorisation menées dans les années 1960.

Que fait-on par rapport à l'utilisation de l'IA ?

- L'image du slide précédent a été réalisée quand les IA génératives n'étaient pas aussi présentes ⇒ Comment faire pour que les 75% de rétention en faisant des exercices pratiques restent vrais ?
- Co-construisons ce slide sachant que [Extraits de « Technopolitique : Comment la technologie fait de nous des petits soldats » de Asma MHALLA] :
 - « Bien des philosophes l'ont dit avant moi : ne pas maîtriser la technique moderne, c'est s'assurer d'en devenir l'esclave. Observer, comprendre et "apprendre à apprendre" sont les conditions pour ne pas être étranger au monde. » (page 256)
 - « L'objectif des opérations cybercognitives est bien d'attaquer sans répit le premier circuit cognitif dit heuristique, celui des biais, de l'intuition, des émotions, de l'immédiateté de la perception, pour mieux inhiber le second, celui du raisonnement, de la logique, du discernement, celui de l'effort et de l'engagement de la pensée complexe. » (page 126-127)
- Wooclap ⇒
 - [❤️ 8] On utilise l'IA pour comprendre une notion ou une erreur, pas pour écrire le code à notre place ou corriger des erreurs.
 - [❤️ 5] Accepter l'aide de l'IA seulement si on comprends ce qui a été fait et qu'on saurait le refaire.
 - [❤️ 4] Ne pas avoir de code rédigé par l'IA, mais accepter de l'utiliser pour des conseils ou autre.

Déroulement

- 1 séance d'introduction : 30/04
 - 1 exercice à faire pour le 07/05 8h30
- 9 CI de 3h
 - Règle des 5 minutes
 - 1 exercice fil rouge à avancer d'une séance à l'autre (faisable en 2024-2025 ?)
 - 1 extrait de chapitre à lire d'une séance à l'autre (faisable en 2024-2025 ?)
- 1 micro-projet du 29/05 au 18/06
 - 27/05 17h15-17h45 : Sujet
 - 28/05 14h30-16h00 : Présentation des premières idées
 - 02/06 – 17/06 : Soutien sur Discord
 - 19/06 **09h00**-12h15 & **13h30**-16h45 : Soutenances

A propos de la présence aux séances

- La présence aux séances de cette UV, c'est comme le port du casque à vélo : pas obligatoire mais indispensable (cf. [ce site](#)).
- À chaque séance, feuille de présence avec « Question du jour » pour que **nous** profitons de l'intelligence collective du groupe.
 - Réfléchissez à des idées de sujet !
 - NB : Le contenu de la feuille de présence sera reporté dans l'application de présence de la DF.

A propos de « *Beaucoup de pratique* »

- « L'important, ce n'est pas la destination, c'est le voyage. » (Robert Louis Stevenson)
- Nous avons besoin de vous pour nous aider à estimer le travail nécessaire pour faire les exercices (et s'imprégner de toutes les notions qu'ils introduisent).
- ⇒ [Fichier Excel partagé](#) à remplir pour indiquer le temps (en minutes) passé sur chacun des exercices C++
 - ⇒ L'an prochain, des statistiques plus précises sur les temps nécessaires
- Merci d'avance
 - D'y indiquer **sincèrement** les temps passés
 - D'y écrire des « SOS » pour qu'on vienne vous aider
 - Autre manière de dire « SOS » = Changer votre rôle Discord en « EtudiantQuestion »

Recommandations

- Quel état d'esprit adopter vis-à-vis de ce cours ?
 - "La phrase la plus excitante à entendre, celle qui annonce de nouvelles découvertes, n'est pas « Eureka », mais plutôt « Tiens, c'est marrant... »" (Isaac Asimov)
- Suivez **tous** les conseils que vous donnent SonarQube et, si vous êtes sous Clion, Clang-Tidy.
 - Un classique : `static_cast<float>(variable_de_type_double)`
- Usez et abusez du salon Discord
 - en soumettant des questions...
 - ...mais aussi en apportant des réponses !
- Ne vous dites pas (et, *a fortiori*, ne dites pas à Loïc ou Michel) que vous n'avez jamais vu des pointeurs, de l'UML, des tests unitaires, des patrons de conception, des traitements d'erreur (exception)...
 - Loïc et Michel connaissent CSC4102 et CSC4103 ;-)

Annales du micro-projet

- 2015 : « La vie, c'est comme une bicyclette, il faut avancer pour ne pas perdre l'équilibre. » (Albert Einstein)
- 2016 : « Si tu veux vraiment l'accomplir, tu trouveras le moyen. Si tu ne veux pas vraiment, tu trouveras l'excuse. » (Jim Rohn)
- 2017 : Carte tirée du jeu de plateau *Dixit*
- 2018 : « |00| »
- 2019 : « Entre le fort et le faible, entre le riche et le pauvre, entre le maître et le serviteur, c'est la liberté qui opprime et la loi qui affranchit » (Henri LACORDAIRE)
- 2020 : « Développez un jeu s'appuyant sur un épisode de la mythologie grecque. »
- 2021 : « Science sans conscience n'est que ruine de l'âme. » (Rabelais)
- 2022 : « Développez un jeu inspiré d'un livre de Jules Verne. »
- 2023 : « 3 Monsieur Madame ont été tirés au hasard pour votre binôme/monôme. Votre jeu doit s'inspirer de 2 de ces 3 Monsieur-Madame, en prenant en compte les noms des deux Monsieur-Madame. Dit autrement, cette inspiration peut aussi s'inspirer de l'ensemble du titre des ouvrages (sans toutefois omettre les noms des Monsieur-Madame concerné·e·s). »
- 2024 : « Votre jeu devra intégrer cet audio (enregistrement d'un rire de bébé). »
- 2025 : ?



Évaluations

- Industriels/Enseignants-chercheurs → Étudiant·es (NB : pas profs / élèves !)
 - Micro-projet
 - Beaucoup de temps hors présentiel
 - Occasion de consolider vos connaissances en C++
 - Occasion de vous former à SFML (ou Unreal)
 - Occasion de vous former à ImGui (ou pas)
 - Occasion d'ajouter une ligne à votre CV
- Étudiant·es → Industriels/Enseignants-chercheurs
 - Fiche d'évaluation sur Moodle
 - Réunion bilan à la fin des oraux de projet
- Occasion de voir, pour chacun·e, si la VAP JIN lui correspond (cf. zoom en slide suivant)

Est-ce que JIN est fait pour moi ? Suis-je fait•e pour JIN ?

1. Abordé en CSC4526

1. Écrire du C++ moderne
2. Design patterns
3. Games programming patterns
4. Tirer le meilleur d'une machine

2. Abordé en CSC4508

1. Multi-threading, gestion de la concurrence
2. Intérieur d'un système d'exploitation
3. Tirer le meilleur d'une machine



Un sujet que CSC4526/JIN4 ne fait qu'effleurer : La Conception Orientée Data

- Programmation Orientée Objet (POO, enseignée à TSP)
 - La POO tend à regrouper des données hétérogènes dans des objets (instances de classes), d'une manière plus simple à appréhender pour le développeur, mais moins efficace pour l'ordinateur.
 - Ces objets sont ensuite manipulés par des instructions dont peu utilisent l'ensemble des données (attributs) qu'ils contiennent.
 - Les caches CPU sont donc souvent pollués par des données inutiles → Plus d'appels à la RAM (plus de cache misses) que nécessaire, et donc une perte de temps significative.
- Versus Conception Orientée Data
 - La Conception Orientée Data est une technique d'optimisation basée sur la mémoire cache des processeurs (CPU). Les données, et les données suivantes adjacentes en RAM sont copiées en mémoire cache CPU pour un accès rapide.
 - La Conception Orientée Data cherche à tirer parti du mécanisme de prefetch de manière optimale. Il s'agit, en général, de stocker les données de plusieurs objets, manipulés les uns à la suite des autres par un certain type d'instruction, de manière contiguë – ou proche, tout du moins – en RAM.
- (Merci à Gibril Günder, JIN 2021-2022, et à son Directeur de stage)

CSC4526/JIN4 ne suffit pas

- Design Patterns : <https://refactoring.guru/design-patterns>
- Montez en puissance en algorithmie
 - Livre « Cracking the Coding Interview, 6th Edition: 189 Programming Questions and Solutions » de Gayle Laakmann McDowell (disponible à la médiathèque)
 - <https://www.hackerrank.com/>, <https://www.codingame.com>, <https://projecteuler.net/>
 - Livre « Programmation efficace : les 128 algorithmes qu'il faut avoir compris et codés en Python au cours de sa vie », Christoph Dürr et Jill-Jênn Vi (disponible à la médiathèque)
- C++
 - J'insiste : Suivre les conseils de SonarQube et Clang-Tidy.
 - Livre de Stroustrup
 - <https://www.learncpp.com/>, <https://github.com/isocpp/CppCoreGuidelines>, <https://cpp.developpez.com/> (Cours + FAQ)
 - <http://en.cppreference.com/w/>
 - <https://www.youtube.com/user/CppCon>, <http://www.bfilipek.com/> (blog d'un fan de C++)

Remarque (avancée) à propos du langage C++

- C++ est un langage normalisé par un comité non dépendant d'éditeurs de logiciels (<https://isocpp.org/>)
- Ce comité normalise une nouvelle version tous les 3 ans.
- Version actuelle = C++23
 - Version utilisée en CSC4526 = C++23
 - NB : Souvent, SonarQube vous fera des suggestions d'améliorations liées à C++23 (sur du code récupéré de StackOverflow, ChatGPT, etc.)
 - FYI
 - En décembre 2024, SFML a basculé en version 3.0.0. Cet *upgrade* de C++03 à C++17 “*is bringing SFML into the world of modern C++!*”
 - Depuis Boost 1.84.0 (13/12/2023), “*C++03 is no longer supported, the minimum required language version is C++11.*”
- Les différents compilateurs n'implémentent pas forcément toutes les fonctionnalités d'une version. En particulier, différences majeures entre Clang et Apple Clang (cf. https://en.cppreference.com/w/cpp/compiler_support/23)

Rien à voir avec le C++, mais occasion assez unique !

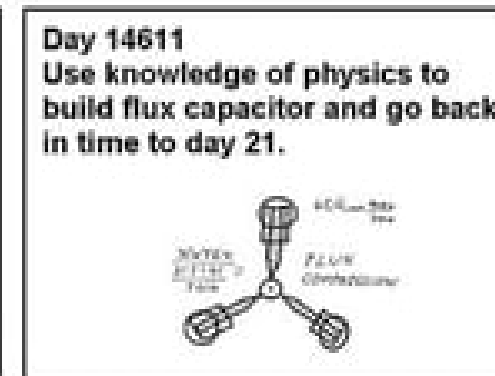
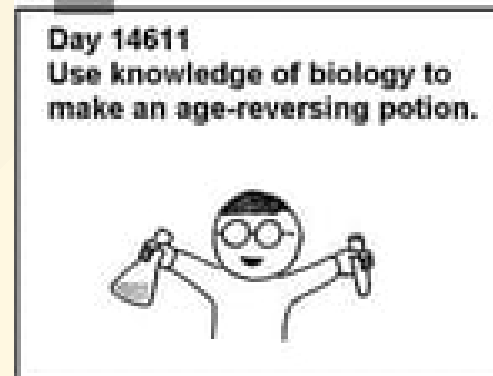
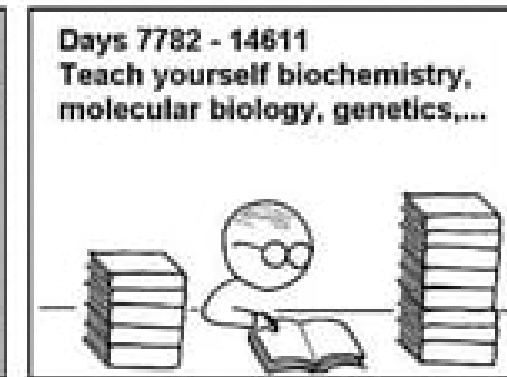
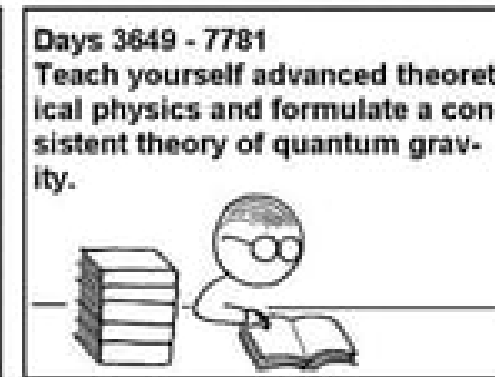
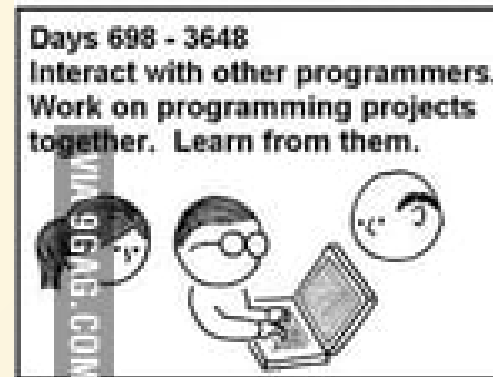
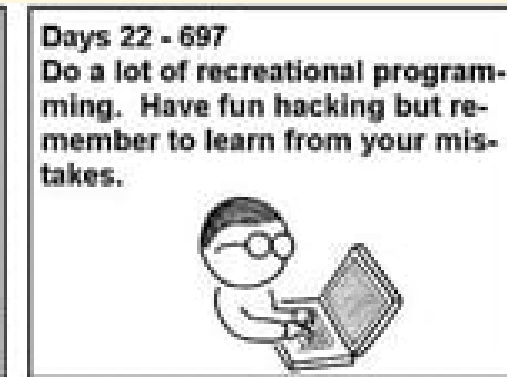
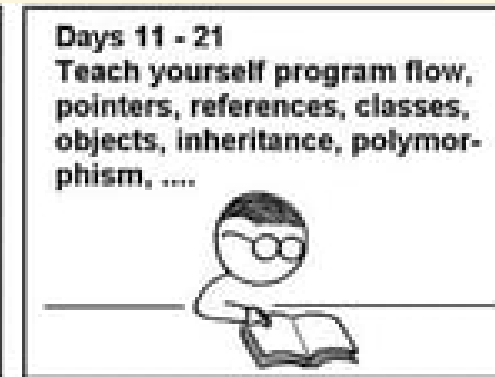
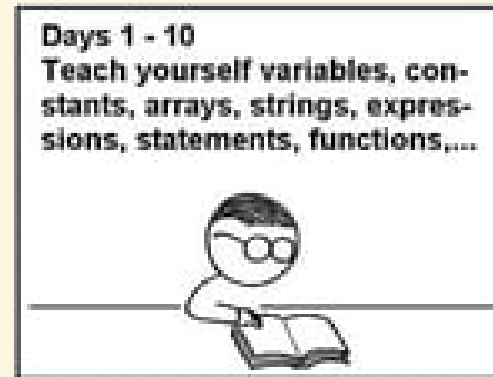
- 10/05/2025
 - 16h30 à 18h00 : Balade « Découverte de la face cachée d'Évry » avec Franck Senaud.
 - 19h30 à ??h?? : Spectacle de Ruiz « 2025 comme possible : Portraits de la jeunesse (Création théâtrale participative) » au Théâtre de l'Agora.
 - Et c'est gratuit !
 - [Plus d'info](#)

Conclusion

1. [Introduction](#)
2. [CSC4526/JIN4 : une UV inscrite dans la VAP JIN](#)
3. [Zoom sur CSC4526/JIN4](#)
4. → [Conclusion](#)

En guise de conclusion

- 30/04-19/06 \Rightarrow 50 jours pour apprendre C++, c'est mieux que 21 jours, mais ça reste peu (cf. gag ci-contre).
- Dans quel état d'esprit êtes-vous : « Ca va être trop, trop bien » ou alors « Aïe, aë, aïe » ?
- En tous cas : (voix suave) « Nous vous souhaitons un agréable voyage sur cette UV en notre compagnie. »



As far as I know, this is the easiest way to "Teach Yourself C++ in 21 Days".
25