

Comment suivre ce cours ?

Pédagogie de ce cours

partir des bonnes pratiques générales, puis voir les spécificités de codage particulières

Gamification : le but des tests n'est pas de sanctionner un apprentissage, mais de permettre à l'apprenant de suivre sa progression. Doit être très pédagogique : beaucoup de petits tests (désacraliser les tests, ne pas voir les tests comme des épreuves), valorisant (tester pas uniquement si acquisition des points difficiles, mais aider à réaliser tout ce que l'on a acquis), revenir régulièrement sur les notions déjà acquises (enseigner, c'est répéter).

Modules : tronc commun d'apprentissage (base du C++, les bonnes pratiques), puis modules optionnels organiser en thèmes : bas niveau/programmation système/embarqué, les interfaces graphiques, les jeux, le C++ old school, allez plus loin, etc.

Qu'est que veut dire "créer une application" ?

Plusieurs approches possibles pour créer un programme :
Plusieurs approches possible. On dev pas pareil pour un micro contrôleur (mémoire, performances) que pour un ordi de bureau ou un serveur de calculs intensifs (performances). Impact sur les méthodes de développement et sur ce que l'on accepte ou non de faire en termes de bonnes pratiques de codage

Création d'un application :

- **Maintenabilité** : corriger les bugs rencontrés durant la vie du programme

- Évolutivité : ajouter de nouvelles fonctionnalités
- Fiabilité : les résultats sont corrects
- Performance : éviter les pertes inutiles de performances.
différence entre chercher la meilleure performance et éviter de faire du code qui explose inutilement les performances. Le second nécessite juste le respect de bonnes pratiques de codage. Le premier nécessite une approche de développement spécifique (profiling)

Donc objectif = arriver le plus rapidement possible au résultat

- KISS = rester simple
- NIH = ne pas réécrire ce que existe déjà
- DRY = ne pas se répéter

Sommaire principal	Chapitre suivant
------------------------------------	----------------------------------

[Cours, C++](#)