

# Classe de maître C++ 14/17

Vous connaissez C++, mais vous avez eu vent que le C++ « moderne » était presque un nouveau langage? Vous aimeriez à la fois apprendre et comprendre cette nouvelle façon de programmer? Suivez cette formation avec Patrice Roy, membre du comité de standardisation du langage.

## Apprendre et comprendre le C++ « moderne ».

Cette classe de maître est dédiée aux programmeurs C++ qui souhaitent une mise à jour de leurs compétences et une meilleure compréhension de ce langage dans son acception moderne.

Les modalités de programmation proposées tiendront du C++ « moderne », donc à partir de [C++11], qui sont radicalement différentes des incarnations précédentes du langage.

La perspective proposée sera résolument « moderne », au sens de C++ 11, et ne véhiculera pas délibérément des pratiques de temps plus anciens.

Cette formation présente en quoi C++14 améliore C++11, puis en abordant les nombreux nouveaux outils mis à notre disposition depuis C++17. Si C++14 est un standard plus humble que son prédécesseur (bien qu'il s'y cache des perles!), C++17 apporte beaucoup de nouveautés, influence nos pratiques de programmation de multiples manières, et contribue directement à la fois la productivité et au plaisir de programmer.

## L'expert Patrice Roy

M. Roy enseigne au CeFTI (Faculté des sciences, Université de Sherbrooke) depuis 2001, et est professeur au Département d'informatique du Collège Lionel-Groulx depuis 1998. Il a été coordonnateur de ce département de 2001 à 2004, et a été coordonnateur du programme Technique de l'informatique de ce collège de 2001 à 2009. Le programme du collège étant coopératif, ses liens avec l'entreprise privée sont nombreux. Il a contribué activement à la conception du programme d'informatique qui fut en vigueur du début des années 2000 jusqu'à 2018 dans ce collège. S'ajoutent à cela quelques tâches d'enseignement au Collège O'Sullivan, à l'ÉTS et pour des groupes d'informatiennes et d'informaticiens en entreprise.

Son passé professionnel récent inclut notamment une participation active au développement et à l'enseignement dans les diplômes de 2e cycle en technologies de l'information (DTI), en génie logiciel (DGL) et en développement du jeu vidéo (DDJV) de l'Université de Sherbrooke. S'ajoutent à cela quelques contributions de type recherche et développement pour des entreprises privées, surtout dans le domaine de l'aérospatiale; trois années à CAE Électronique Ltée, affecté principalement au développement de simulateurs d'hélicoptères militaires pour la marine britannique, en particulier dans l'interface aux instructeurs et la simulation des messages à contenu vocal; une contribution à titre de consultant pour IREQ, pour le projet MONITEQ, un système de maintenance préventive d'appareils électriques gagnant de plusieurs prix et finaliste au R&D100 en 1995. Il a complété un doctorat sur la contextualisation dans un système massivement réparti à l'Université de Sherbrooke sous la direction conjointe du professeur Bessam Abdulrazak et de docteur Yacine Belala, et siège sur plusieurs comités, en particulier WG21 (le comité de standardisation du langage C++).

C++ 14/17

1 000 \$

Bénéficiez de 300 \$ de rabais en remboursement salarial\*

3 et 4 juin, 2019

15 heures

Université de Sherbrooke  
Campus de Longueuil

Formation présentée en français

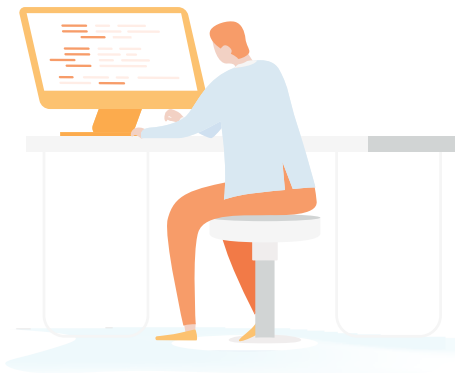
Politique d'inscription

**Vivez l'expérience  
de formation  
TECHNOCompétences.**

300 \$ Remboursé

Bénéficiez de 20\$/ heure de  
remboursement salarial  
sur toutes les formations  
TECHNOCompétences

*Dîner inclus pour chaque journée de formation.  
Le matériel informatique est fourni également.*



Suite →

### Classe de maître C++ 11.

Formation éligible à la loi du 1% + 300\$ de remboursement par participant !

<https://www.cpmt.gouv.qc.ca/grands-dossiers/loi-compétences/index.asp>

fzmezzar@technocompetences.qc.ca

514 840 1237 EXT. 339

# Classe de maître C++ 14/17

Vous connaissez C++, mais vous avez eu vent que le C++ « moderne » était presque un nouveau langage? Vous aimeriez à la fois apprendre et comprendre cette nouvelle façon de programmer? Suivez cette formation avec Patrice Roy, membre du comité de standardisation du langage.

C++ 14/17

1 000\$

Bénéficiez de 300 \$ de rabais en  
remboursement salarial\*

3 et 4 juin, 2019

15 heures

Université de Sherbrooke  
Campus de Longueuil

Formation présentée en français

## Préalable

Cette formation, tenue sur 15 heures est destinée à des programmeuses et programmeurs ayant une expérience préalable avec C++.

Il est présumé que les participantes et les participants ont au préalable intégré les concepts et outils de C++ 11. Ce standard étant vaste et riche, ne pas avoir été suffisamment en contact avec lui constituerait un obstacle à la compréhension de ce que propose la formation sur C++ 14 et C++ 17.

## Survol de la formation

Quelques exercices ponctuent la formation et sont proposés aux participant(e)s; de plus, une plage d'entre deux et trois heures de résolution de problèmes permettant de mettre en application certains outils et concepts présentés au cours de la formation. Une discussion de possibles solutions à ces problèmes clôt le tout.

La formation propose une mise à jour des connaissances des participantes et des participants en vue de tirer profit des concepts et outils propres aux standards C++ 14 [Cpp14Std] et C++ 17 [Cpp17Std]. Notez qu'au moment d'écrire ces lignes, le standard C++ 17 est considéré complet, au sens où il ne devrait plus y avoir de nouveaux ajouts en termes de mécanismes et de fonctionnalités, mais que le processus de standardisation appelle à des commentaires de pays, ce qui est susceptible d'entraîner des changements au texte existants.

Le survol suivant donne un aperçu aérien de la formation proposée. Les éléments en caractères gras sont ceux considérés prioritaires, et seront par conséquent couverts en plus grand détail que les autres; notez toutefois que ces éléments dépendent pour la plupart de concepts préalables, ce qui fait que des sujets moins centraux aux préoccupations des participantes et participants devront aussi être couverts pour faire en sorte que les sujet centraux puissent être présentés de manière compréhensible.

## C++ 11 – a priori

Ce tronçon de 60 à 75 minutes vise à couvrir les principes clés qui sous-tendent la programmation avec depuis C++ 11. L'objectif est de situer C++ dans un contexte dit « moderne » et de permettre aux participantes et aux participants d'exprimer tout besoin quant à un a priori qui pourrait être surprenant.

Les diapositives seront survolées rapidement et laissées aux participant(e)s à titre de document de soutien, sans plus.

Dans ce tronçon, sont survolés :

- `<chrono>`
- `<random>`
- `<tuple>`
- `<type_traits>`
- **auto, decltype et declval()**
- **Expressions  $\lambda$**
- Initialisation standardisée
- Boucle for sur des intervalles
- **Templates variadiques**
- **Sémantique de mouvement**
- **Perfect Forwarding**
- **Références de relais**
- using à titre de template typedefs
- Clause noexcept
- **constexpr**
- **Littéraux maison**
- Opérations emplacement()
- **Pointeurs intelligents**
- `<thread>`

## C++ 14

Principalement reconnu comme un complément à C++11, C++ 14 complète et rend plus agréable encore la programmation dans ce langage.

Les thématiques proposées pour ce tronçon vont comme suit :

- **Expressions  $\lambda$  polymorphiques**
- **Blocs de capture enrichis**
- **Variables templates**
- **Fonctions de type auto**
- Séparateurs d'unités
- Littéraux binaires
- Annotation `[[deprecated]]`
- **Expressions constantes généralisées**
- Conversions contextuelles enrichies
- **Initialisation des membres d'agrégats**
- Verrous partagés
- Chaînes de caractères quoted

Suite →

### Classe de maître C++ 11.

Formation éligible à la loi du 1% + 300\$ de remboursement par participant !

<https://www.cpmt.gouv.qc.ca/grands-dossiers/loi-competes/index.asp>

fzmezzar@technocompetences.qc.ca

514 840 1237 EXT. 339

# Classe de maître C++ 14/17

Vous connaissez C++, mais vous avez eu vent que le C++ « moderne » était presque un nouveau langage? Vous aimeriez à la fois apprendre et comprendre cette nouvelle façon de programmer? Suivez cette formation avec Patrice Roy, membre du comité de standardisation du langage.

- Type `integer_sequence`
- Fonction `std::exchange()`
- Fonction `std::make_unique()`
- Fonction `std::make_reverse_iterator()`
- Fonctions `std::cbegin()` et `std::cend()`
- Fonctions `std::rbegin()` et `std::rend()`
- Littéraux standards pour `std::string`, `std::complex` et `<chrono>`
- Foncteurs « diamants »
- **Alias pour des traits (suffixes `_t`)**
- Nouveaux traits : `std::is_final`, `std::is_nullptr`
- Accéder à un élément d'un `std::tuple` par son type (si cela n'est pas ambigu)
- Algorithmes sur la base de deux intervalles
- Fouille hétérogène dans un `std::map` et dans un `std::set`
- Fonctions de hachage standards pour des énumérations
- Libération dynamique de mémoire avec conscience de la taille du bloc à libérer, etc.

Ce tronçon se conclut sur un atelier de résolution collective de problèmes, où les participant(e)s sont placé(e)s face à un problème concret de design et de programmation, problème qui doit être résolu à partir des acquis de la formation. Les équipes comparent par la suite leur solution, le formateur propose la sienne, et un échange entre experts s'ensuit sur les impacts des décisions de design prises de part et d'autres.

## C++ 17

C++ 17 pourrait être qualifié de « drôle de standard ». La communauté attendait de ce jalon des révolutions, qui ne sont pas venues : les gros morceaux attendus, en particulier les modules, les concepts et les coroutines, sont pour l'essentiel livrés, mais sous forme de spécifications techniques plutôt que sous forme officiellement standardisée. L'intention est d'expérimenter pour mieux comprendre l'impact de ces nouveaux mécanismes.

En retour, C++ 17 est une mise à jour considérable sur le plan des outils, et qui comprend quelques surprises agréables. C++ 17 est une mise à jour importante pour tout ce qui a trait à la productivité des programmeuses et des programmeurs, de même qu'un outil précieux pour ce qui touche au plaisir de programmer. De plus, certains mécanismes de ce standard surprennent par leur puissance et par leur polyvalence.

C++ 14/17

1 000\$

Bénéficiez de 300 \$ de rabais en remboursement salarial\*

3 et 4 juin, 2019

15 heures

Université de Sherbrooke  
Campus de Longueuil

Formation présentée en français

Les thématiques proposées pour ce tronçon sont :

- Allègement de `static_assert`
- `std::uncaught_exceptions`
- Annoter des énumérations
- `namespace A::B::C;`
- `0 + ... + args` et autres Fold Expressions
- `__has_include`
- `[[fallthrough]]`, `[[nodiscard]]`, `[[maybe_unused]]`
- Des  $\lambda$  constexpr (en partie)
- Répétitives for avec sentinelles
- `if(decl;cond)`
- `switch(decl;cond)`
- Littéraux à virgule flottante hexadécimaux
- `enum class` en tant que types propres
- Guides de déduction, pour fins d'allègement syntaxique de la construction des templates (éventuelle disparition de `make_pair()` et autres)
- `if constexpr` (ceci peut être gros)
- Clarifier l'ordre d'évaluation des expressions
- `template <auto N>`
- `std::launder()`
- Amélioration des opérations d'insertion dans les conteneurs associatifs
- `void_t` (The C++ Detection Idiom)
- `invoke()`
- Affectation déstructurante (Structured Bindings)
- Le Class Type Argument Deduction (CTAD) et les guides de déduction
- `floor()`, `ceil()`, `round()`, `abs()` sur des types de `<chrono>`
- `as_const()`
- Un `lock_guard<>` variadique : `scoped_lock`
- Algorithmes parallèles
- Fonctions mathématiques spéciales de `<cmath>` (autrefois `<tgmath>`)
- `any`
- `optional<T>`
- `string_view`
- Nouveaux algorithmes de recherche de sous-chaînes
- `apply(f,tup)`
- `shared_ptr<T[]>`
- `sample()`
- `not_fn()`
- `is_always_lock_free`
- Détection constexpr de la taille des Cache Line
- `clamp()`
- `hypot(x,y,z)`
- `is_contiguous_layout`
- Versions constexpr de `reverse_iterator`, `array`, `addressof()`, etc.
- `shared_ptr::weak_type`
- `gcd()`
- Conversions de nombre à string (et inversement) à haute vitesse
- `variant<T0,T1,...>`
- `std::visit()`
- Allocateurs polymorphiques, etc.

Encore une fois, ce tronçon se conclut sur un atelier de résolution collective de problèmes, où les participant(e)s sont placé(e)s face à un problème concret de design et de programmation, problème qui doit être résolu à partir des acquis de la formation. Les équipes comparent par la suite leur solution, le formateur propose la sienne, et un échange entre experts s'ensuit sur les impacts des décisions de design prises de part et d'autres.

### Classe de maître C++ 11.

Formation éligible à la lois du 1% + 300\$ de remboursement par participant !

<https://www.cpmt.gouv.qc.ca/grands-dossiers/loi-competes/index.asp>

fzmezzar@technocompetences.qc.ca

514 840 1237 EXT. 339