AJC FORMATION

CYCLE

20 days - 140 hours

ADVANCED C ++

AJC FORMATION - 6 rue Rougemont 75009 - 01 81 51 64 85 – formonsnous@ajc-formation.fr
S.A.R.L. au capital de 10 000 € - SIRET : 50096702100030
N° de déclaration d'activité 117542701 75 (ce numéro ne vaut pas agrément de l'Etat)

www.ajc-formation.fr
www.ajc-classroom.fr

PROGRAM OF THE TRAINING CYCLE

AJC FORMATION – 6 rue Rougemont 75009 - 01 81 51 64 85 – formonsnous@ajc-formation.fr
S.A.R.L. au capital de 10 000 € - SIRET : 50096702100030
N° de déclaration d'activité 117542701 75 (ce numéro ne vaut pas agrément de l'Etat)

www.ajc-formation.fr
www.ajc-classroom.fr

TRAINING CYCLE ADVANCED C++

Program

GOALS

- C ++ reminders.
- Explore the new features of the different versions of C ++ over time
- Put into practice

Teaching methods. For all trainees, the course will include the following:

- Alternating exercises, practical cases, multiple choice questions and theoretical notions
- Reviews

Teaching resources

- AJC provides each trainee with access to our remote platform as well as any useful software as part of each module
- The course materials will be delivered via our download platform Quest and / or AJC Classroom

TEACHING METHODS AND RESOURCES

Information about virtual classes

- For training in virtual class, with @JC CLASSROOM, you will benefit from the same possibilities and interactions with your trainer as during a face-to-face training: your training will take place in continuous connection 7/7.
- You will be able to interact directly with the trainer and the teaching team through our videoconferencing system, but also thanks to the forums and chats in @JC CLASSROOM.
- Your trainer will be able to check the progress of your work and estimates your level using exercises and practical cases. This will allow him to provide you with educational follow-up and personalized advice throughout the duration of the training.
- Our technical team will send you the connection details (access, identifiers, dates, times and hotline number) by email as soon as you register.
- If you encounter a connection problem, you can reach our technical assistance hotline at 01 82 83 72 41 or by email at any time (before or even during the training) (hotline@ajc-formation.fr)

PRE-REQUISITE

- The learner must have knowledge of C ++

PARTICIPANTS

- Consultants, Engineers, Developers,....

LIEU

Distance

CERTIFICATE

Training certificate

Program — List of modules

TOOLS	SOURCE MANAGEMENT WITH GIT	2 days
C ++ DEVELOPMENT	C ++ REMINDERS	2 days
	ADVANCED C ++ PROGRAMMING AND NEW VERSIONS 11,14,17, 20	11 days
DESIGN	DESIGN PATTERNS C ++	2 days
PROJECT	FINAL PROJECT	3 days

DETAILED PROGRAMS

TOOLS

AJC FORMATION

ADVANCED C++

SOURCE MANAGEMENT WITH GIT

2 days,
14 hours

DISTANCIAL

PROGRAMME DU MODULE

Version control

- Why version its source code?
- Basic concepts of version control

The DVCS principle (Ditributed Control version)

- What does decentralization bring?
- Principle of operation
- Branch, depot, merge, rebase and all DVCS concepts
- Version control

Day-to-day use

- Create / clone a repository
- View the status of the work tree
- View changes
- Save Changes
- Browse revision history
- Find the author of a modification
- Basic concepts of version control

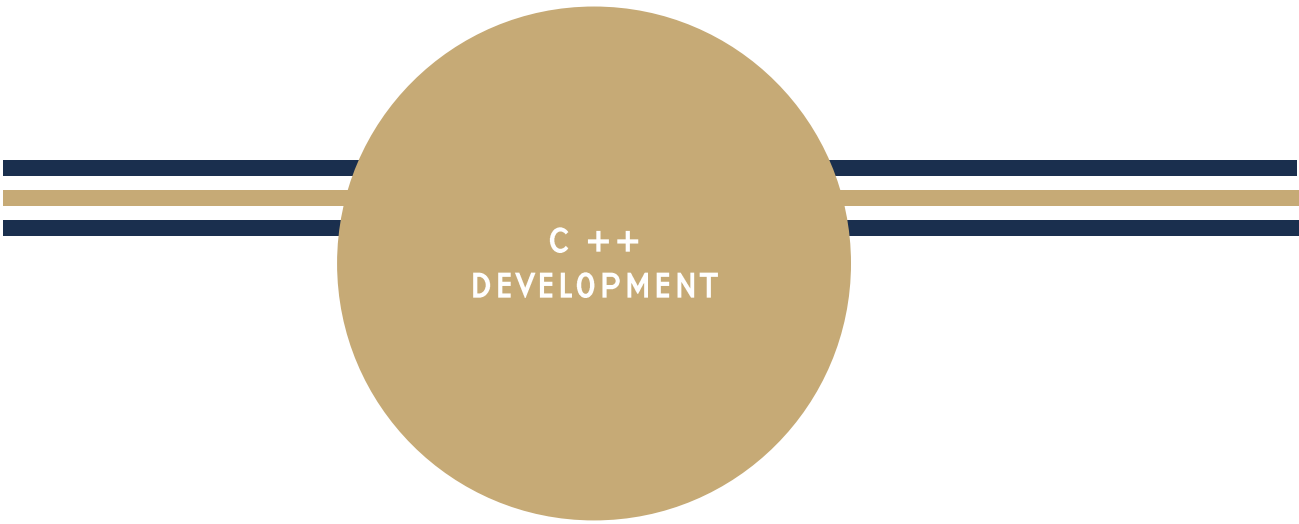
Warehouse and branch management

- Create a branch
- Go from branch to branch with merges or rebase
- Update a repository
- Export your repository
- Remote repositories

GOALS

- Understanding the DVCS principles
- Learn to manage your source code with Git
- Learn to collaborate with Git repositories
- Know how to handle the tools annexed to Git

DETAILED PROGRAM


C ++
DEVELOPMENT

ADVANCED C++

C ++ REMINDERS

2 days,
14 hours

DISTANCIAL

PROGRAMME DU MODULE

Object Oriented Programming

Namespaces

Preprocessor

Pointers

Smart pointers

Pointers and Reference

Templates

Multithreading

Best practices and design rules in C ++

Avoid side effects

Stack & heap

Management of compilation parameters

The basics of contract programming

GOALS

- Remind the basic concepts
- Know good practices and design rules in C ++

DETAILED PROGRAM

AJC FORMATION

ADVANCED C++

ADVANCED C ++ PROGRAMMING AND NEW VERSIONS 11,14,17, 20

11 days,
77 hours

DISTANCIAL

PROGRAMME DU MODULE

Presentation

- Standards 11, 14 and 17
- Support and activation in the main compilers

Deepening C ++ 03

- SFINAE
- RVO
- Smart pointer
- STL algorithms

News & Changes in C ++ 11

- Language peculiarities
  - alignof, alignas, References rvalue constexpr
  - Type Alias
  - Standard attributes
- New on the constants
  - nullptr
  - constexpr
  - Character strings / string
  - User type
  - Typed enumeration
- Generalization of begin/ end and extended for loop
- Initialization lists & unified initialization
- New on types
  - Keywords auto and decltype
  - New function syntax
  - Type Aliases and templates
- New generic types
  - std ::array
  - list unidirectional
  - tuples
  - hash tables
  - smart Pointer
- Static assertions
- Lambda functions and closures
- New in the classes
  - Keywords override, final, delete, default
  - move constructor/assignment, delegate)
- Variadic template
- Function attributes, types and variables
- Multithreading support
  - thread
  - mutex
  - condition
  - atomic
- Support for regular expressions
- Random number management

C ++ 14 changes

- Automatic deduction of function returns
- decltype(auto)
- Binary numbers constants
- Digital separator
- Generic lambda
- Initialization by aggregate
- More flexibility in constexpr
- Template for variables
- Attribute deprecated
- Access to tuples by type (get←→)
- User-defined literals (s, h, min, i, etc.)

GOALS

- Master programming in C ++
- Know the different evolutions of C ++
- Put into practice

DETAILED PROGRAM

ADVANCED C++

ADVANCED C ++ PROGRAMMING AND NEW VERSIONS 11,14,17, 20 (Following)

11 days,
77 hours

DISTANCIAL

PROGRAMME DU MODULE

Changes made by C ++ 17

- Definition of namespace simplified
- The use of typename instead of class for templates
- New attributes
- Utf8 support
- Auto and initialization list
- Tests at compile time
- Initialization in if and switch
- "Structured binding "
- "Template deduction"for builders
- Improved lambda support
- New types
  - Std : string_view, compile-time constants, ...
  - std :: variant
  - std :: byte
  - std ::optional and std ::any
  - std :: filesystem
- Parallelization of the stl
- Various additions to the library

Changes made by C ++ 20

- Named initializers
- Operator spaceship
- Consteval/constinit
- Span and ranges
- Concepts and constraints
- Modules
- Coroutines

GOALS

- Master programming in C ++
- Know the different evolutions of C ++
- Put into practice

DETAILED PROGRAM

DESIGN

AJC FORMATION

ADVANCED C++

DESIGN AND DESIGN PATTERNS

2 days,
14 hours

DISTANCIAL

PROGRAMME DU MODULE

Presentation of the design

- Reminder of the fundamental concepts of OOP and UML programming.
- UML notation diagrams. Its contributions for the design.
- Design issues.
- Reuse by inheritance.
- Fundamentals of Object Design
- The evolution strategy with the opening / closing principle (OCP).
- The principle of substitution of Liskov (LSP).
- The concept of polymorphism.
- The impact of Object design on the life cycle of projects.

Principles of organization in packages

- The package as a design unit.
- Principles of delivery / reuse equivalence (REP) and common reuse (CRP).
- The breakdown of packages thanks to the principle of common closure (CCP).
- The organization between packages: principles of acyclic dependence (ADP) and dependence / stability relationship (SDP).

Principles of building classes

- Rational management of dependencies with dependency reversal (DIP).
- Reduction of apparent complexity through separation of interfaces (ISP).
- The distribution of responsibilities with the GRASP principle.

Principles of Design Patterns

- The technical principles of designing an Object application.
- Origin and scope of patterns.
- The advantages and limits of Design Patterns.
- Solve recurring problems and ensure the sustainability of developments.

The founding patterns of Gamma and GoF

- The catalog of patterns of the "band of four".
- Objectives and benefits.
- Isolate the creation of objects from their use.
- Refine the allocation of responsibilities through behavioral patterns.
- Improve class structuring.

GOALS

- Understand the fundamental principles of Object Design
- Apply the basic rules for dividing an application into a package
- Apply the principles of construction of the classes of an application
- Learn to implement the main Design Patterns

DETAILED PROGRAM

PROJECT

AJC FORMATION

ADVANCED C++

FINAL PROJECT

2 days,
14 hours

DISTANCIAL

PROGRAMME DU MODULE

Course of the module

- Trainees work independently, in pairs. They are free to make the appropriate choices, to develop the parts they deem most necessary and to provide their own solutions to the problems posed.
- The trainer supervises the trainees by his presence and answers questions. He intervenes to support a binomial in difficulty or to take stock of the whole group on unearned notions. He may need to help to increase or complete some knowledge.

GOALS

- Apply the learning outcomes by completing the mini projects carried out throughout the training cycle

AJC FORMATION

CONTACT US

AJC FORMATION
01 81 51 64 85
formonsnous@ajc-formation.fr
6 rue ROUGEMONT
75009 PARIS

www.ajc-formation.fr
www.ajc-classroom.fr