

C C++

Filière





PROGRAMME DE LA FILIERE

Programme

OBJECTIFS

- Acquérir les bases de la programmation structurée en langage C.
- Acquérir les bases de la programmation Objet en C++.
- Mises en pratique

Méthodes pédagogiques. Pour l'ensemble des stagiaires, le cours intégrera les suivantes :

- Alternance d'exercices, cas pratiques, QCM et de notions théoriques
- Evaluations

Moyens pédagogiques

- AJC met à la disposition de chaque stagiaire un accès à notre plateforme à distance ainsi qu'éventuellement les logiciels utiles dans le cadre de chaque module
- Les supports de cours seront remis via notre la plate-forme de téléchargement Quest et/ou AJC Classroom

METHODES ET MOYENS PEDAGOGIQUES

Informations concernant les classes virtuelles

- Pour les formations en classe virtuelle, avec @JC CLASSROOM, vous profiterez des mêmes possibilités et interactions avec votre formateur que lors d'une formation présentielle : votre formation se déroulera en connexion continue 7h/7.
- Vous pourrez échanger directement avec le formateur et l'équipe pédagogique à travers notre système de visioconférence, mais aussi grâce aux forums et chats présents dans @JC CLASSROOM.
- Votre formateur sera à même de vérifier l'avancement de votre travail et de vous évaluer à l'aide d'exercices et de cas pratiques. Cela lui permettra de vous apporter un suivi pédagogique et des conseils personnalisés pendant toute la durée de la formation.
- Notre équipe technique vous enverra les modalités de connexion (accès, identifiants, dates, heures et numéro de la hotline) par mail dès votre inscription.
- Si vous rencontrez un problème de connexion, vous pourrez joindre à tout moment (avant ou même pendant la formation) notre hotline assistance technique au 01 82 83 72 41 ou par mail (hotline@ajc-formation.fr)

PRE-REQUIS

- L'apprenant doit avoir des notions d'Algorithmie

PARTICIPANTS

- Consultants, Ingénieurs, Développeurs, ...

LIEU

Distanciel

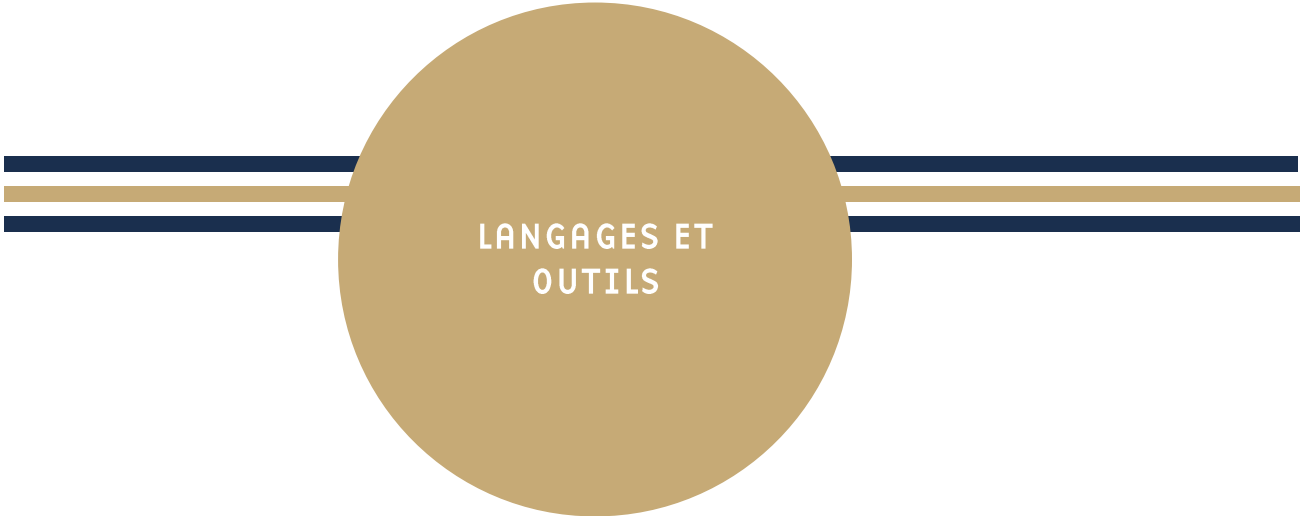
CERTIFICATION / ATTESTATION

Attestation de formation

Programme - Contenu pédagogique

LANGAGES ET OUTILS	ALGORITHMIE	2 jours
	GESTION DES SOURCES AVEC GIT	2 jours
	LANGAGE C	3 jours
PROJET	PROJET C	1 jour
LANGAGES ET OUTILS	PROGRAMMATION C++	10 jours
PROJET	PROJET FINAL	2 jours

PROGRAMMES
DÉTAILLÉS



LANGAGES ET
OUTILS



ALGORITHMIE

PROGRAMME DU MODULE

Introduction à l'algorithmie

Instructions de Base

- Variables
- Affectation
- Tests
- Boucles
- Exercices

Procédures et fonctions Introduction

Définition d'une procédure

Définition d'une fonction

Appel d'une procédure

Appel de fonction et retour

Maîtriser les notions de fichiers

Acquérir les bases des méthodes de programmation structurée nécessaires à l'apprentissage de tout langage de programmation

Apprendre à raisonner sur un algorithme

Exemple : Les tris à bulle, les tris par inversion ...

Découvrir et mettre en œuvre la traduction d'un algorithme dans un langage de programmation

OBJECTIFS

- Présenter les principes fondamentaux de la programmation et de l'algorithmique et expliquer les notions communes à tous les langages de programmation
- S'approprier les structures logiques et la démarche de résolution d'un problème de façon structurée et indépendante de toute contrainte matérielle ou logicielle
- Résoudre des problèmes plus ou moins complexes



GESTION DES SOURCES AVEC GIT

PROGRAMME DU MODULE

Le contrôle de version

- Pourquoi versionner son code source ?
- Les concepts de base du contrôle de version

Le principe DVCS (Distributed version Control)

- Qu'apporte la décentralisation ?
- Principe de fonctionnement
- Branche, dépôt, merge, rebase et tous les concepts DVCS
- Le contrôle de version

Utilisation au jour le jour

- Créer/cloner un dépôt
- Consulter l'état de l'arbre de travail
- Visualiser les modifications
- Enregistrer les modifications
- Parcourir l'historique des révisions
- Retrouver l'auteur d'une modifications
- Les concepts de base du contrôle de version

Gestion de dépôt et branches

- Créer une branche
- Passer de branche en branche avec les merges ou le rebase
- Mettre à jour un dépôt
- Exporter son dépôt
- Les dépôts distants

OBJECTIFS

- Comprendre les principes DVCS
- Apprendre à gérer son code source avec Git
- Apprendre à collaborer avec les dépôts Git
- Savoir manipuler les outils annexes à Git



LANGAGE C

PROGRAMME DU MODULE

Introduction

- Introduction
 - Présentation du langage C
 - Évolution du C et du C++
 - Principe de compilation
 - Compilation des programmes
 - Écriture d'un programme
- Outils de développement
 - Les compilateurs C
 - Environnements de développement
 - Installation de Code::Block
 - Création d'un nouveau projet
- Premier programme
 - Ouvrir un projet C
 - Le premier programme
 - Source du premier programme

Éléments

- Vocabulaire
 - Caractères autorisés
 - Les identificateurs
 - Séparateurs
- Blocs d'instruction
 - Les mots-clefs
 - Blocs d'instructions
 - Les commentaires
 - Programme main
- Déclarations
 - Préprocesseur directive `#include`
 - Variables
 - Portée et initialisation des variables

Types

- Types de base
 - Type caractères : ``char``
 - Type entier : ``int``
 - Les flottants : ``float``
 - Déclaration des variables

Constance

- Constantes

- Déclaration de variable constante

Opérateurs

- Opérateurs simples
- Opérateurs d'affectation ``=``
- Opérateurs arithmétiques
- Les conversions numériques implicites
- Les opérateurs relationnels

Autres opérateurs

- Les opérateurs logiques
- Les opérateurs de manipulation de bits
- Les opérateurs d'incrément et de décrément
- Affectation composée
- Opérateur virgule

Opérateurs avancés

- Opérateur conditionnel ternaire
- Les opérateurs de cast
- L'opérateur `&`
- L'opérateur `sizeof`
- Priorité des opérateurs (ordre décroissant)

Contrôles

- Instructions conditionnelles
 - `if---else`
 - `switch`
- Branchement non conditionnel `break`
- Branchement non conditionnel continue
- Branchements non conditionnels `goto`

Les pointeurs

- Importance des pointeurs
- Pointeurs et gestion mémoire
- Adresse et valeur
- Types de pointeurs : variables, fonctions

OBJECTIFS

- Savoir développer en langage C
- Connaître les aspects avancés du langage C.
- Comprendre l'intérêt des pointeurs



PROGRAMMATION C++

PROGRAMME DU MODULE

Programmation Orientée Objet

- Modélisation, Abstraction
- Encapsulation
- Classe et Objets
- Instanciation d'objet
- Les membres, attributs, méthodes
- Héritage, Polymorphisme

Du C au C++

- Préambule
- Définitions, Types
- Paramètres de fonctions par référence
- Fonctions inline
- Appels de fonctions C
- Entrées et sorties simples

Les classes

- Les classes
- Déclaration dans le fichier cpp
- Méthodes, Paramètres par défaut
- Masquage, Surchage
- Les constructeurs

Constructeurs

- Exemples d'utilisation
- Types de constructeurs
- Constructeur par copie
- Constructeurs par transtypage
- Usage du mot clé explicit
- Constructeurs à arguments multiples
- Liste d'initialisations, Le destructeur

Qualifieurs

- Constance
- Variables statiques

- Méthodes statiques
- Autres types de classe de stockage
- Constance
- Fonctions amis, Classes amies

Les exceptions

- Exemple d'exception
- Remarques
- Attraper une exception
- La classe exception et sa dérivée stdexcept
- Réponses
- Laisser échapper une exception
- Déclaration de fonction avec exception non traitées
- Faire/défaire

Espaces de nommage

- Présentation
- Déclaration
- Définition
- Utilisation
- Exemples connus
- Exclusion de méthodes d'un namespace
- Espace de nommage anonyme

Préprocesseur

- Les directives
- Directives include et define simple
- Instructions conditionnelles
- ifndef define ifdef
- identificateur
- Les macros
- Macros prédéfinies

OBJECTIFS

- Appréhender les concepts objets (Classe, Encapsulation, polymorphisme, héritage) qui ne sont pas présents en langage C.
- Mise en pratique à l'aide de nombreux exercices
- Prendre en main l'interface de développement Eclipse
- Mettre en oeuvre des différents outils Eclipse pour développer des applications C : édition, compilation, debug
- Etre capable de surcharger des opérateurs et des méthodes pour redéfinir leur exécution.
- Connaître bonnes pratiques et règles de conception en C++



PROGRAMMATION C++ (Suite 1)

PROGRAMME DU MODULE

Dérivation - Héritage

- Classe fille
- Déclaration de la classe fille
- Encapsulation
- Particularité amusante
- Héritage multiple
- Portée d'héritage
- Redéfinition vs surcharge
- Problématiques des classes dérivées

Polymorphisme

- Méthodes virtuelles
- Contrainte sur les fonctions virtuelles
- Fonctions virtuelles pures et classes abstraites

Pointeurs

- Rappels
- Définition d'un pointeur
- Difficultés de notation
- Pointeurs et allocation mémoire
- Désallocation
- Organisation de la mémoire
- Pointeurs et classes
- Le pointeur `*this*`
- Arithmétique sur les pointeurs
- Cas d'utilisation des pointeurs : pas

RTTI

- Définition
- Exemple pratique
- Règles à respecter
- `typeid`
- Utilisation

Pointeurs intelligents

- Problématique des pointeurs
- Présentation des pointeurs intelligents
- `unique_ptr`
- Exemple de `unique_ptr`
- `shared_ptr`
- Exemple de `shared_ptr`
- `weak_ptr`
- Avantages et inconvénients des pointeurs intelligents

OBJECTIFS

- Appréhender les concepts objets (Classe, Encapsulation, polymorphisme, héritage) qui ne sont pas présents en langage C.
- Mise en pratique à l'aide de nombreux exercices
- Prendre en main l'interface de développement Eclipse
- Mettre en oeuvre des différents outils Eclipse pour développer des applications C : édition, compilation, debug
- Être capable de surcharger des opérateurs et des méthodes pour redéfinir leur exécution.
- Connaître bonnes pratiques et règles de conception en C++



PROGRAMMATION C++ (Suite 2)

PROGRAMME DU MODULE

C11

- Présentation
- Templates variadiques
- Templates externes
- Assertions statiques
- Expressions et fonctions lambda
- Sémantique RValue Reference/Move
- Enumérations typées
- Le type tuple
- Support des expressions régulières
- Tables de hachage
- Boucles d'intervalle

Multithreading

- Principes
- Démarrage et détachement d'un thread
- La classe `std::call_once`
- L'espace de noms `this_thread`
- Futures / promises et `packaged_task`
- Programmation asynchrone avec `async`
- Politiques de démarrage
- Comparaison `thread` versus `future`
- Partage de ressources et mécanismes de synchronisation
- Mutexes et données atomiques
- `Unique-lock` et `lock_guard`
- Introduction à la programmation Multicore

ECLIPSE

- Introduction

- Les objectifs et les principes d'Eclipse.
- Les concepts de base : vue, éditeur, perspective, espace de travail, projet, répertoire, fichier, nature.
- Le démarrage d'Eclipse. Fonctions d'aide. Gestion des vues et des perspectives. Gestion des ressources.
- Fonctions de navigation, de recherche. Gestion des projets.
- Le debug
- La perspective Debugger dans Eclipse.
- Les différents Breakpoints et Watchpoints.
- L'inspection des variables ou expressions, la navigation dans la pile d'exécution.
- Le contrôle de l'exécution. Le debug réparti.
- Gestion des versions avec GIT
- La configuration d'Eclipse
- La gestion des préférences. La gestion des propriétés.
- Les références entre projets. L'ajout de Plug-ins et Features. Le paramétrage d'Eclipse.
- La configuration d'Eclipse
- La gestion des préférences. La gestion des propriétés.
- Les références entre projets. L'ajout de Plug-ins et Features. Le paramétrage d'Eclipse.

OBJECTIFS

- Appréhender les concepts objets (Classe, Encapsulation, polymorphisme, héritage) qui ne sont pas présents en langage C.
- Mise en pratique à l'aide de nombreux exercices
- Prendre en main l'interface de développement Eclipse
- Mettre en oeuvre des différents outils Eclipse pour développer des applications C : édition, compilation, debug
- Etre capable de surcharger des opérateurs et des méthodes pour redéfinir leur exécution.
- Connaître bonnes pratiques et règles de conception en C++



PROGRAMMATION C++ (Suite 3)

PROGRAMME DU MODULE

Surcharge des opérateurs

- Liste des opérateurs qui peuvent être surchargés
- Surcharge d'un opérateur par une fonction membre
- Surcharge d'un opérateur par une fonction non membre
- Surcharge de l'opérateur d'affectation `operator=`
- Opérateur de conversion vers un autre type
- Symétrie

Surcharge de méthodes

- Templates
- Présentation
- Patrons de fonction
- Utilisation
- Remarques
- Piège sur les pointeurs
- Patron de fonction à l'intérieur d'une classe
- Instanciation implicite et explicite
- Spécialisation
- Patrons de classe
- Exemple
- Fonctions exportées
- Template sur les opérateurs
- Foncteurs
- A quoi servent les foncteurs ?
- Cas d'utilisation
- Performance

Les bonnes pratiques et règles de conception en C++

- Organiser votre projet travailler en

itération

- Quelles seront les étapes de création ?
- Quelles technologies vais-je utiliser ?
- Laquelle sera la mieux adaptée à mon projet ?
- Quelle est la structure du projet à adopter ?
- Quels outils seront nécessaires ?
- Faire du versioning
- Développer, commenter
 - Respectez les conventions
 - Structure
 - Nommage
 - Les design patterns
 - L'optimisation de code
- Factoriser
- Profiling/Profilage de code
- Réaliser une veille régulière
- Editer des rapports de bug
- Quelques concepts de base
 - Écrire un code compréhensible
 - Une déclaration par ligne
 - Choisissez des noms explicites
 - Une ligne = une instruction
 - Respectez des règles d'indentation strictes
 - Respectez les conventions
 - Les mêmes règles pour tout le projet
 - Préférez les énumérations
 - Préférez les tableaux à une dimension

OBJECTIFS

- Appréhender les concepts objets (Classe, Encapsulation, polymorphisme, héritage) qui ne sont pas présents en langage C.
- Mise en pratique à l'aide de nombreux exercices
- Prendre en main l'interface de développement Eclipse
- Mettre en oeuvre des différents outils Eclipse pour développer des applications C : édition, compilation, debug
- Être capable de surcharger des opérateurs et des méthodes pour redéfinir leur exécution.
- Connaître bonnes pratiques et règles de conception en C++



PROGRAMMATION C++ (Suite 4)

PROGRAMME DU MODULE

Du bon usage des commentaires

- Un commentaire ne doit pas paraphraser le code
- Un commentaire ne sert pas à marquer les étapes
- Les commentaires à visée pédagogique
- Les commentaires sous forme de cartouche .

Éviter les effets de bord

La pile et le tas

Gestion des paramètres de compilation

Les bases de la programmation par contrat

Problèmes d'instances

- Éviter la création des objets
 - maîtriser et réduire le nombre d'objets
 - Ré utiliser et factoriser les instances
- Utiliser les « Factories » pour optimiser la création des objets
- Introduction de « singletons » dans une application :Avantages et inconvénients
- Gestion d'algorithmes de pools d'objets
- Introduction des caches de ressources
- Introduction des caches de calculs
- Renoncer aux habitudes du C Un

tableau d'éléments ? C'est std::vector

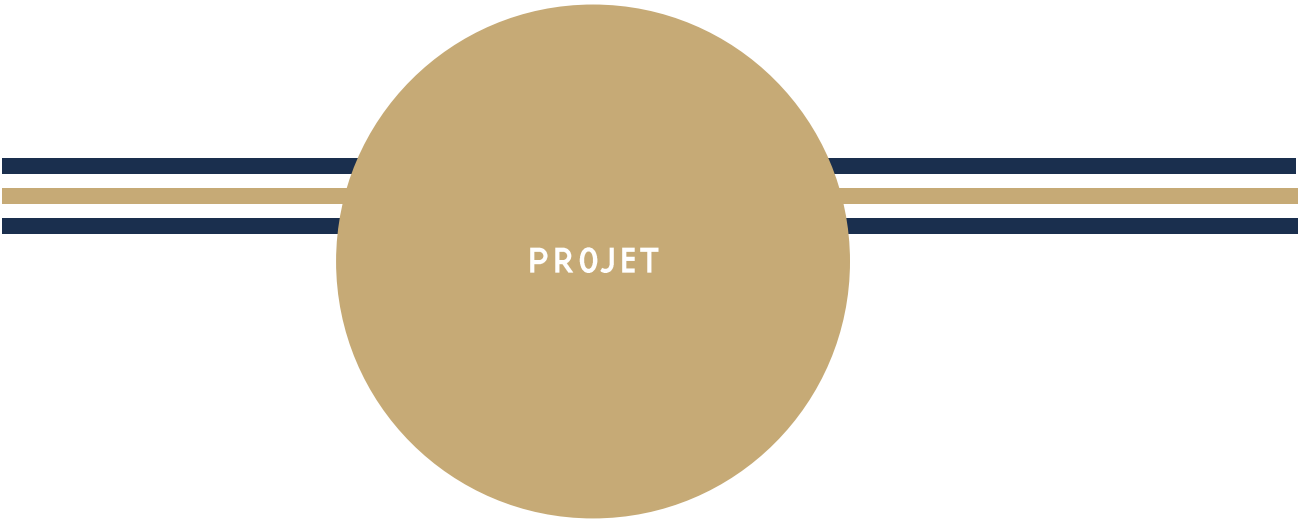
- Une chaîne de caractères ? C'est std::string
- Le flux d'entrée par défaut ? C'est std::cin
- Utiliser un fichier ? C'est std::fstream
- Convertir ses données avec std::stringstream
- Transtypages C style

Pointeurs et référence

- Démystifier les pointeurs
- Qu'est-ce qu'une référence ?
- Quelle différence entre les deux ?
- Transmettre des paramètres : par valeur ou par référence ?
- Transmettre des paramètres, par référence ou par pointeur ?
- Rvalue-reference ?

OBJECTIFS

- Appréhender les concepts objets (Classe, Encapsulation, polymorphisme, héritage) qui ne sont pas présents en langage C.
- Mise en pratique à l'aide de nombreux exercices
- Prendre en main l'interface de développement Eclipse
- Mettre en oeuvre des différents outils Eclipse pour développer des applications C : édition, compilation, debug
- Être capable de surcharger des opérateurs et des méthodes pour redéfinir leur exécution.
- Connaître bonnes pratiques et règles de conception en C++





C C++

PROJET C

1 jour,
7 heures



DISTANCIEL

PROGRAMME DU MODULE

Déroulement du module

- Les stagiaires travaillent en toute autonomie, en binôme. Ils sont libres d'effectuer les choix adaptés, de développer les parties dont ils jugent avoir le plus besoin et d'apporter leurs propres solutions aux problèmes posés.
- Le formateur encadre les stagiaires par sa présence et répond aux questions. Il intervient pour épauler un binôme en difficulté ou pour faire le point à l'ensemble du groupe sur des notions non acquises. Il peut être amené à approfondir ou compléter certaines connaissances.

OBJECTIFS

- Mettre en application les acquis de la formation sur le langage C

PROGRAMME DETAILLE



C C++

PROJET FINAL

PROGRAMME DU MODULE

Déroulement du module

- Les stagiaires travaillent en toute autonomie, en binôme. Ils sont libres d'effectuer les choix adaptés, de développer les parties dont ils jugent avoir le plus besoin et d'apporter leurs propres solutions aux problèmes posés.
- Le formateur encadre les stagiaires par sa présence et répond aux questions. Il intervient pour épauler un binôme en difficulté ou pour faire le point à l'ensemble du groupe sur des notions non acquises. Il peut être amené à approfondir ou compléter certaines connaissances.

2 jours,
14 heures



DISTANCIEL

OBJECTIFS

- Mettre en application les acquis de la formation en complétant les mini projets réalisés dans tout le cursus

PROGRAMME DETAILLE

NOUS CONTACTER

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



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

