Durée :	5 jours soit 35 heures	Référence :	IF-LANGC

Public visé :

Cette formation s'adresse aux développeurs et informaticiens qui souhaitent acquérir les bases du langage C, qu'ils découvrent ce langage ou qu'ils aient besoin de consolider des connaissances acquises de façon empirique. Elle convient notamment :

- aux développeurs venant d'un autre langage (Java, Python, C#, PHP, etc.) amenés à évoluer vers le C dans le cadre d'un projet système, embarqué ou industriel
- aux informaticiens devant lire, maintenir ou faire évoluer une base de code C existante
- aux étudiants, alternants ou jeunes diplômés en informatique souhaitant consolider leurs bases avant une prise de poste technique
- aux profils techniques d'administrations, de collectivités ou d'entreprises privées confrontés à des environnements logiciels reposant sur le C (systèmes, automatismes, outillage interne)

Pré-requis :

- Avoir déjà programmé dans au moins un langage (notion de variable, de boucle, de fonction, de test conditionnel), quel qu'il soit
- Être à l'aise avec l'usage d'un terminal / d'une ligne de commande
- Aucune connaissance préalable du langage C n'est requise

Objectifs pédagogiques :

À l'issue de la formation, le stagiaire sera capable de :

- Mettre en œuvre une chaîne de compilation C (préprocesseur, compilation, édition de liens) avec GCC ou Clang
- Écrire des programmes C structurés en respectant la syntaxe et les conventions du langage
- Utiliser les opérateurs, les structures conditionnelles et les boucles pour construire des algorithmes
- Déclarer, appeler et organiser des fonctions au sein d'un projet réparti sur plusieurs fichiers sources
- Manipuler les chaînes de caractères et les structures de données du C de façon sécurisée
- Utiliser les pointeurs et gérer dynamiquement la mémoire en appliquant les bonnes pratiques de prévention des erreurs mémoire
- Lire et écrire des données dans des fichiers, et mettre en œuvre une structure de données chaînée simple

Compétences acquises à l'issue de la formation :

- Être capable de configurer et d'utiliser une chaîne de compilation C (GCC/Clang, options de compilation, débogueur) pour développer et diagnostiquer un programme
- Être capable d'écrire des programmes C structurés en mobilisant correctement les types de données, les opérateurs et les structures de contrôle du langage
- Savoir concevoir et organiser des fonctions, et structurer un projet en plusieurs fichiers sources compilés séparément
- Maîtriser la manipulation sécurisée des chaînes de caractères et des structures de données (struct, union) du C
- Être capable d'utiliser les pointeurs et de gérer dynamiquement la mémoire en appliquant les bonnes pratiques de prévention des fuites et des erreurs mémoire
- Être en mesure de mettre en œuvre la persistance de données dans un fichier et de construire une structure de données chaînée pour un cas d'usage concret

Modalités pédagogiques :

Session dispensée en présentiel ou téléprésentiel, selon la modalité inter-entreprises ou intra-entreprises sur mesure. La formation est animée par un(e) formateur(trice) durant toute la durée de la session et présentant une suite de modules théoriques clôturés par des ateliers pratiques validant l'acquisition des connaissances. Les ateliers peuvent être accompagnés de Quizz. L'animateur(trice) présente la partie théorique à l'aide de support de présentation, d'animation réalisée sur un environnement de démonstration. En présentiel comme en téléprésentiel, l'animateur(trice) accompagne les participants durant la réalisation des ateliers.

Note relative au programme : Les ateliers, démonstrations, travaux pratiques et scénarios présentés dans ce programme sont

fournis à titre d'exemples illustratifs et ne sont pas exhaustifs. Afin de garantir une adéquation permanente avec les évolutions technologiques, le contenu pédagogique ainsi que les environnements de travaux pratiques (TP) sont susceptibles d'être adaptés, ajustés ou mis à jour au cours du temps.

Moyens et supports pédagogiques :

Cadre présentiel

Salles de formation équipées et accessibles aux personnes à mobilité réduite.

- Un poste de travail par participant
- Un support de cours numérique ou papier (au choix)
- Un bloc-notes + stylo
- Vidéoprojection
- Connexion Internet
- Accès extranet pour partage de documents et émargement électronique

Cadre téléprésentiel

Session dispensée via notre solution iClassRoom s'appuyant sur Microsoft Teams.

- Un compte Office 365 par participant
- Un poste virtuel par participant
- Un support numérique (PDF ou Web)
- Accès extranet pour partage de documents et émargement électronique

Modalités d'évaluation et de suivi :

Avant

Afin de valider le choix d'un programme de formation, une évaluation des prérequis est réalisée à l'aide d'un questionnaire en ligne ou lors d'un échange avec le formateur(trice), qui valide la base de connaissances nécessaires.

Les stagiaires disposent également, depuis leur extranet, d'un **outil d'auto-positionnement** leur permettant d'évaluer leur niveau de maîtrise des compétences visées par la formation. Cette auto-évaluation est réalisée avant le démarrage de la formation à l'aide d'un curseur de positionnement allant du niveau le plus faible au niveau le plus élevé.

Pendant

Après chaque module théorique, un ou plusieurs ateliers pratiques permettent de valider progressivement l'acquisition des connaissances. Un quiz peut accompagner les ateliers pratiques afin de mesurer la compréhension des notions abordées.

Après

À l'issue de la formation, les stagiaires réalisent une nouvelle **auto-évaluation** sur les mêmes compétences que celles évaluées avant la formation. Cette comparaison permet de mesurer leur perception de la progression réalisée et de visualiser les compétences développées au cours de la formation.

Lorsque le programme de formation le prévoit, un examen de certification est organisé conformément aux modalités définies par l'éditeur ou le centre de certification (TOSA, Pearson VUE, ENI, PeopleCert…).

Le formateur réalise également une **évaluation individuelle des compétences** de chaque stagiaire sur l'ensemble des compétences visées par la formation. Pour chacune d'elles, il indique si la compétence est :

- Acquise ;
- En cours d'acquisition ;
- Non acquise.

Lorsque la compétence est évaluée comme **en cours d'acquisition** ou **non acquise**, le formateur formule des observations précisant les difficultés rencontrées ainsi que les actions ou recommandations permettant au stagiaire de poursuivre sa montée en compétences.

Enfin

Un questionnaire de satisfaction permet au participant d'évaluer la qualité de la prestation de formation.

Informations sur l'accessibilité :

Nos formations sont, dans la mesure du possible, conçues pour être accessibles à toutes et à tous. Afin de garantir les meilleures

conditions d'accueil et d'apprentissage pour les personnes en situation de handicap, nous vous invitons à contacter notre référente handicap certifiée :
Céline SOLATGES – 05 61 34 39 80 – csolatges@iform.fr
Nous vous remercions de bien vouloir nous communiquer toute information utile à ce sujet en amont de la formation, afin de mettre en place les adaptations nécessaires et d'assurer un accompagnement optimal.
Pour en savoir plus sur les dispositifs d'accompagnement existants, vous pouvez consulter les sites suivants :

- AGEFIPH
- FIPHFP
- MDPH de votre département

Description / Contenu

Module 1 – Environnement de développement et chaîne de compilation C

- Panorama des normes du langage : C89, C99, C11, C17, C23 ; norme de référence utilisée en formation (C17) et principales évolutions apportées par C23
- Compilateurs GCC et Clang, phases de la compilation (préprocesseur, compilation, assemblage, édition de liens)
- Structure d'un programme C : fichier source, fonction main, en-têtes

Démonstration : compilation en ligne de commande avec les options -std=c17 -Wall -Wextra -Wpedantic -O0 -g, lecture des messages du compilateur, premier pas à pas dans un débogueur (gdb)
Exemples de travaux pratiques :

- Écriture et compilation d'un premier programme C
- Correction d'un programme à partir des avertissements du compilateur

Module 2 – Types de données et structuration de l'information

- Variables de types prédéfinis et types à largeur fixe (stdint.h)
- Tableaux, constantes et énumérations
- Structures (struct), unions et typedef

Exemples de travaux pratiques :

- Exercice de déclaration de variables de différents types (simples, tableaux)
- Lancement du projet fil rouge : modélisation d'un contact (structure regroupant nom, prénom, téléphone, e-mail) et création d'un tableau statique de contacts

Module 3 – Opérateurs et structures de contrôle

- Opérateurs arithmétiques, relationnels, logiques et bit à bit ; affectations composées
- Conversions et promotions de types
- Structures conditionnelles (if/else, switch) et boucles (for, while, do...while)
- Directives de préprocesseur utiles en pratique : #define, compilation conditionnelle, include guards

Atelier pratique : lecture et refactoring d'un programme fourni comportant des structures de contrôle imbriquées
Exemples de travaux pratiques :

- Ajout au projet fil rouge d'un menu interactif (saisir, afficher, quitter) permettant de peupler et consulter le tableau de contacts
- Validation du format d'un numéro de téléphone ou d'une adresse e-mail saisis par l'utilisateur

Module 4 – Fonctions et organisation d'un projet C

- Déclaration, définition et appel de fonctions ; prototypes
- Portée des variables (locale, globale, statique) et passage de paramètres par valeur
- Découpage d'un programme en plusieurs fichiers sources et fichiers d'en-tête
- Compilation séparée et introduction au Makefile

Cas concret : refactorisation du projet fil rouge en fonctions dédiées (ajout, recherche, affichage d'un contact)
Exemples de travaux pratiques :

- Répartition du projet fil rouge en plusieurs fichiers sources et compilation à l'aide d'un Makefile

Module 5 – Chaînes de caractères et pointeurs

- Manipulation des chaînes de caractères : strlen, strcmp, strncpy, strncat, snprintf ; fonctions de conversion (strtol, strtod) ; fonctions à proscrire (gets, strcpy, strcat, sprintf) et raisons de leur mise à l'écart

- Notion de pointeur : adresse et valeur, arithmétique de pointeurs, lien entre pointeurs et tableaux, pointeurs de fonctions

Démonstration : observation en débogueur des adresses mémoire et du contenu des pointeurs
Exemples de travaux pratiques :

- Sécurisation de la saisie et de la comparaison des champs texte (nom, e-mail) du projet fil rouge à l'aide des fonctions de la bibliothèque standard
- Parcours du tableau de contacts à l'aide de pointeurs

Module 6 – Gestion dynamique de la mémoire et manipulation de fichiers

- Allocation dynamique : malloc, calloc, realloc, free ; bonnes pratiques (vérification des retours, mise à NULL après libération, prévention des fuites et des doubles libérations)
- Ouverture, lecture, écriture et fermeture de fichiers ; gestion des erreurs d'entrée/sortie

Démonstration : détection, à l'aide de Valgrind ou d'AddressSanitizer, d'une fuite mémoire introduite volontairement dans une variante du projet fil rouge
Exemples de travaux pratiques :

- Migration du tableau statique de contacts vers une allocation dynamique (realloc) permettant un annuaire de taille variable
- Sauvegarde et rechargement de l'annuaire de contacts dans un fichier texte

Module 7 – Structure de données chaînée : projet de synthèse

- Principe de la liste chaînée simple : ajout, parcours, suppression, libération complète de la structure

Mise en situation professionnelle : migration de l'annuaire de contacts vers une liste chaînée pour une gestion dynamique de l'ajout et de la suppression ; revue de code collective à partir d'une grille de bonnes pratiques (gestion mémoire, robustesse, lisibilité)
Exemple de travaux pratiques :

- Projet final : finalisation de l'annuaire de contacts sous forme de liste chaînée (ajout, affichage, recherche, suppression, libération complète de la mémoire)