Stage pratique de 5 jour(s)
Réf : LGC

Participants

Développeurs, ingénieurs, chefs de projets proches du développement.

Pré-requis

Connaissances de base en programmation.

Prix 2017 : 2610€ HT

Dates des sessions

Paris
10 avr. 2017, 19 juin 2017
21 aoû. 2017, 16 oct. 2017
18 déc. 2017

Aix
19 juin 2017, 25 sep. 2017
27 nov. 2017

Bordeaux
10 avr. 2017, 26 juin 2017
11 sep. 2017, 11 déc. 2017

Bruxelles
12 juin 2017, 18 sep. 2017
11 déc. 2017

Geneve
12 juin 2017, 18 sep. 2017
11 déc. 2017

Grenoble
10 avr. 2017, 26 juin 2017
11 sep. 2017, 11 déc. 2017

Lille
19 juin 2017, 25 sep. 2017
27 nov. 2017

Luxembourg
12 juin 2017, 18 sep. 2017
11 déc. 2017

Lyon
10 avr. 2017, 26 juin 2017
11 sep. 2017, 11 déc. 2017

Montpellier
19 juin 2017, 25 sep. 2017
27 nov. 2017

Nantes
19 juin 2017, 25 sep. 2017
27 nov. 2017

Rennes
19 juin 2017, 25 sep. 2017
27 nov. 2017

Sophia-antipolis
19 juin 2017, 25 sep. 2017
27 nov. 2017

Strasbourg
19 juin 2017, 25 sep. 2017
27 nov. 2017

Toulouse
10 avr. 2017, 26 juin 2017
11 sep. 2017, 11 déc. 2017

Modalités d'évaluation

L'évaluation des acquis se fait tout au long de la session au travers des multiples

Programmation en C

Best

Ce stage intensif vous permettra d'acquérir une connaissance réellement opérationnelle du langage. Il vous expliquera le fonctionnement des différents mécanismes et vous montrera leur mise en œuvre grâce à de nombreux exercices pratiques. A l'issue de ce stage, vous serez en mesure d'écrire des programmes C robustes et portables.

OBJECTIFS PEDAGOGIQUES

Maîtriser la chaîne de production d'un programme écrit en langage C
Mettre en œuvre les opérateurs, les expressions et les structures de contrôle du langage C
Manipuler des structures de données, des tableaux, des pointeurs et des chaînes de caractères
Organiser le code d'un programme à l'aide de fonctions.
Exploiter les principales bibliothèques standard du langage C

1) Premiers pas en C
2) Opérateurs et expressions
3) Structures de contrôle
4) Tableaux, pointeurs et chaînes de caractères
5) Les structures
6) Les fonctions
7) Compilation séparée, classe d'allocation
8) Le préprocesseur
9) Les bibliothèques standard

Travaux pratiques

Des machines sous système Unix ou Windows (PC) seront mises à la disposition des participants de manière à mettre en pratique les notions présentées.

1) Premiers pas en C

- Présentation du langage C, ses atouts.
- Le C++ par rapport au C. Normes C++11 et C11.
- Les fichiers sources (.c, .h).
- Structure générale d'un programme.
- La syntaxe de base du langage.
- Les types de données et les constantes de base.
- Variables globales et locales.
- Stockage et passage de paramètres.
- Entrées/sorties formatées.
- Les commentaires.
- Utilisation élémentaire de la chaîne de production.
- Les environnements d'édition, de compilation et d'exécution.
- Exécution d'un premier programme.

2) Opérateurs et expressions

- Opérateurs arithmétiques.
- Mécanismes d'évaluation des expressions.
- Post et pré-incrémentation de décrémentation.
- Précédence et associativité des opérateurs.
- Opérateurs d'affectation.
- Mécanismes de fonctionnement des expressions logiques.
- Expressions logiques dans les instructions while, if...
- Opérateurs de comparaison : <, >, ==, !=...
- Opérateurs logiques : ET, OU, négation.
- Les types numériques composés. Règle de conversion dans les expressions mixtes. Conversions implicites/explicites.
- Initialisation des variables.
- Arithmétique sur les adresses.
- Formats d'entrée/sortie associés aux types numériques.
- Opérateurs bit à bit : ET, OU, OU exclusif, complément à 1, négation. Opérateurs de décalage : >>, <<.
- Expression conditionnelle avec l'opérateur ternaire.

Travaux pratiques
Mise en œuvre des opérateurs et expressions.

3) Structures de contrôle

- Notion de blocs.
- Les structures de boucles : while, for.
- Instructions de contrôle de boucles : break, continue.
- Structures de choix : if, else, else if.
- Structure de choix multiple : switch.

exercices à réaliser (50 à 70% du temps).

Compétences du formateur

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

Moyens pédagogiques et techniques

• Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.

• A l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.

• Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

Travaux pratiques
Mise en œuvre des structures de contrôle.

4) Tableaux, pointeurs et chaînes de caractères

- Définition, initialisation et accès aux éléments d'un tableau.
- Définition d'un pointeur. Récupérer l'adresse mémoire d'un objet. Accéder au contenu d'un pointeur.
- Equivalences pointeurs/tableaux.
- Calculs sur les pointeurs.
- Chaînes de caractères.
- Exemples de manipulation de chaînes de caractères.
- Les chaînes de caractères Unicode de C11.

Travaux pratiques
Manipulation de tableaux, de pointeurs et des chaînes de caractères.

5) Les structures

- Intérêts des structures.
- Déclarer, initialiser et accéder aux champs d'une structure.
- Utiliser des structures imbriquées.
- Créer de nouveaux types en utilisant Typedef.
- Les champs de bits.
- Les unions.
- Les énumérations.
- Les structures et énumérations anonymes de C11.
- Définir des pointeurs sur structures.

Travaux pratiques
Implémentation de nouvelles structures de données.

6) Les fonctions

- Définition d'une fonction.
- Appel d'une fonction.
- Passage de paramètres : par valeur ou par référence.
- Code retour d'une fonction. Les types de retour.
- La fonction "main".

Travaux pratiques
Découper son code à l'aide de fonctions. Gérer les appels de fonctions.

7) Compilation séparée, classe d'allocation

- Mécanisme de fonctionnement de la chaîne de production.
- Utilisation de bibliothèque de sources.
- Notion de Makefile.
- Configuration mémoire d'un programme C (pile, tas...).
- Classes d'allocation des variables (auto, register, static, extern).
- Différents cas de figure de la compilation séparée.
- Notion d'objet externe.
- Cas des données globales et statiques.
- Cas des données locales.
- Règle de visibilité.
- Compléments sur les fonctions et les initialisations.

8) Le préprocesseur

- Utilisation des macros prédéfinies (constantes symboliques). Définir ses propres macros avec #define.
- Définir des macros comme des fonctions. Utilisation des marqueurs # et ##.
- Annuler la définition de constante avec #undef.
- La compilation conditionnelle : #if, #ifdef, #ifndef, #elif, #endif.
- Inclure des ressources avec #include.

Travaux pratiques
Utilisation des directives du préprocesseur. Mise en place de la compilation conditionnelle.

9) Les bibliothèques standard

- Les fonctions de calcul mathématique (sqrt, sin...).
- Les fonctions d'entrées/sorties (fprintf, fscanf...).
- Les fonctions d'accès aux fichiers (fread, fwrite...).
- Les fonctions de manipulation de chaînes de caractères (strlen, strcat...).
- Les fonctions de gestion de la mémoire (malloc, free...).
- Mise en place de structures chaînées (listes chaînées, arbres n-aire...).
- Les fonctions "sécurisées" de la librairie standard C11 (strcat_s, strlen_s, ...).

Travaux pratiques
Utilisation des principales fonctions des bibliothèques standard.