**L**a différence clé entre C et C++ est que C est un langage de programmation procédural et ne prend pas en charge les classes et les objets, alors que C++ est une combinaison du langage de programmation procédural et orienté objet. C++ peut donc être appelé un langage hybride. Le tableau suivant présente la différence entre C et C++ en détail.

Table de comparaison

| **C** | **C++** |
| --- | --- |
| C a été développé par Dennis Ritchie entre 1969 et 1973 chez AT & T Bell Labs. | C++ a été développé par Bjarne Stroustrup en 1979 avec le prédécesseur de C++. |
| Comparé à C++, C est un sous-ensemble de C++. | C++ est un sur-ensemble de C. C++ peut exécuter la plupart du code C, alors que C ne peut pas exécuter de code C ++. |
| C ne supporte pas la programmation orientée objet; par conséquent, il ne prend pas en charge le polymorphisme, l’encapsulation et l’héritage. | En tant que langage de programmation orienté objet, C++ prend en charge le polymorphisme, l’encapsulation et l’héritage. |
| En C (parce que c’est un langage de programmation procédural), les données et les fonctions sont des entités séparées et libres. | En C++ (lorsqu’il est utilisé comme langage de programmation orienté objet), les données et les fonctions sont encapsulées ensemble sous la forme d’un objet. Pour la création d’objets, la classe fournit un plan de structure de l’objet. |
| En C, les données sont des entités libres et peuvent être manipulées par un code extérieur. En effet, C ne prend pas en charge l’encapsulation d’informations. | En C++, Encapsulation masque les données pour garantir que les structures de données et les opérateurs sont utilisés comme prévu. |
| C étant un langage de programmation procédurale, c’est un langage basé sur les fonctions. | Alors que C++, étant un langage de programmation orientée objet, c’est un langage orienté objet. |
| C ne prend pas en charge la surcharge des fonctions et des opérateurs. | C++ supporte à la fois surcharge des fonctions et d’opérateurs. |
| C ne permet pas de définir des fonctions dans des structures. | En C++, les fonctions peuvent être utilisées dans une structure. |
| C n’a pas de fonctionnalité de NAMESPACE | C++ utilise NAMESPACE pour éviter les conflits de noms de variables/fonctions/classes… |
| C utilise des fonctions pour les entrées/sorties. Par exemple, scanf et printf. | C++ utilise des objets pour la sortie. Par exemple cin et cout. |
| C ne supporte pas les variables de référence. | C++ supporte les variables de référence. |
| C n’a pas de support pour les fonctions virtuelles et friend. | C++ prend en charge les fonctions virtuelles et friend. |
| C fournit les fonctions malloc() et calloc() pour l’allocation dynamique de la mémoire et free() pour la dés-allocation de mémoire. | C++ fournit un nouvel opérateur pour l’allocation de mémoire et l’opérateur de suppression pour la dés-allocation de mémoire. |
| C ne fournit pas de support direct pour le traitement des erreurs (également appelé traitement des exceptions) | C++ fournit un support pour la gestion des exceptions. Les exceptions sont utilisées pour des erreurs « difficiles » qui rendent le code incorrect. |
| Extension du fichier C est .C | Extension du fichier C++ est .CPP |

Points importants sur C/C++

* C++ est dérivé de C.
* « C with classes » était le nom précédent de C++
* Un certain nombre de programmes C peuvent être facilement compilés dans le compilateur C++
* Tous les opérateurs C sont valables en C++
* Les expressions C++ et C sont identiques

###### Conclusion

Comprendre la différence entre C et C++ et leurs forces peut aider les entreprises à les exploiter plus efficacement dans leurs projets de développement logiciel. Cette connaissance pourrait contribuer au succès de vos futures projets qui seront basés sur C/C++.