

**L'intro**

# C++

Jesse Liberty

- ▶ **Concevez et développez une application en C++**
- ▶ **Travaillez avec les variables, les constantes, les instructions et les expressions pour construire un programme**
- ▶ **Maîtrisez la conception orientée objet**



Tous les codes sources  
téléchargeables gratuitement  
sur [www.campuspress.fr](http://www.campuspress.fr)



CAMPUSPRESS

-005-326-1

2-005-326-1

**L'intro**

# C++



*Jesse Liberty*

Retrouvez  
tous nos livres sur  
[www.campuspress.fr](http://www.campuspress.fr)



**CAMPUSPRESS**



# Table des matières

<b>I. Introduction</b> .....	1
L'apprentissage du C++ ne doit pas être nécessairement difficile .....	1
Compiler le code .....	2
Conventions de ce livre .....	2
Fractionner le code .....	3
<b>1. Introduction au C++</b> .....	5
Les programmes .....	6
Résoudre les problèmes .....	7
La programmation procédurale, structurée et orientée objet .....	7
Pourquoi le C++ ? .....	9
Pourquoi la programmation orientée objet ? .....	9
Gérer la complexité .....	10
Les objets .....	10
L'encapsulation .....	11
La délégation .....	11
La spécialisation .....	13
La généralisation et le polymorphisme .....	13
Les trois piliers .....	14
Comment se font l'analyse et la conception orientées objet .....	14
L'analyse et la conception orientées objet pour les petits projets .....	16
La vision .....	17
Analyser les objectifs .....	18
Une conception "vite fait mal fait" .....	19
L'implémentation .....	20
Les programmes et le code source .....	20
Les compilateurs .....	21
Votre environnement de développement .....	21
L'éditeur de texte .....	21
Compiler le code source .....	22
Créer un fichier exécutable à l'aide de l'éditeur de liens .....	23
Le cycle de développement .....	23
Decryptix.cpp : votre premier programme en C++ .....	25
Les erreurs de compilation .....	26
La livraison .....	27
Les étapes suivantes .....	27



<b>2. Mise en route</b> .....	29
Quand un projet peut-il être qualifié de petit ? .....	30
Pourquoi étudier un processus valable seulement pour les petits projets ? .....	30
Se lancer à l'eau .....	30
Créer le projet .....	30
Examiner le code .....	32
Analyser le code .....	34
Les espaces de nom .....	35
using namespace std .....	35
Au cœur du code .....	37
Renvoyer une valeur .....	38
La fonction main() .....	39
Utiliser cout pour afficher à l'écran .....	39
Les variables .....	41
Les constantes .....	50
<b>3. Flux du programme</b> .....	55
Construire robuste .....	55
Qu'essayez-vous de réaliser ? .....	58
Résoudre les problèmes avec des boucles .....	58
Les opérateurs relationnels .....	59
Les blocs d'instructions et les instructions composées .....	60
Les opérateurs logiques .....	61
L'instruction if .....	62
Court-circuiter l'évaluation .....	64
La priorité des opérateurs relationnels .....	64
Récapitulons .....	65
La boucle do while .....	65
Les constantes énumérées .....	66
Retour au code .....	67
Comment obtenir une réponse booléenne de l'utilisateur .....	68
L'opérateur d'égalité == .....	69
else .....	69
L'opérateur conditionnel, ou ternaire .....	70
Récapitulons .....	71
<b>4. Création des classes</b> .....	73
Pourquoi des classes ? .....	73
Créer de nouveaux types : les classes .....	74
L'interface et l'implémentation .....	75
La notion de client .....	75
Examiner le code .....	76
Déclarer la classe .....	76



Les classes et les objets .....	76
Les variables membres .....	77
Les fonctions et méthodes membres .....	77
La taille des objets .....	78
Les fichiers .....	78
Les constructeurs .....	78
Les destructeurs .....	79
Implémenter les méthodes .....	79
Inclure le fichier d'en-tête .....	81
Implémenter le constructeur .....	82
L'initialisation .....	82
Le débogueur .....	84
Examiner le constructeur .....	85
Les autres méthodes .....	85
Stocker la solution .....	85
Qu'est-ce qu'un tableau ? .....	86
Initialiser les tableaux .....	87
Les éléments d'un tableau .....	88
Ecrire au-delà de la fin du tableau .....	89
Générer la solution .....	90
Examiner le fichier DefinedValues .....	94
<b>5. Exécution du jeu .....</b>	<b>97</b>
L'implémentation en ligne .....	98
Les méthodes membres constantes .....	100
La signature .....	101
Passer des paramètres par référence ou par valeur .....	102
Les références et le passage par référence .....	106
Les pointeurs .....	107
Qu'est-ce qu'un pointeur ? .....	107
Les adresses mémoire .....	108
Déréférencer .....	112
Pour y voir clair dans les opérateurs .....	113
Les tableaux .....	114
Les tableaux en tant que pointeurs .....	115
Passer le tableau comme pointeur .....	118
Travailler avec ASSERT .....	121
Comment fonctionne ASSERT .....	121
Explorer le programme ligne par ligne .....	127
<b>6. Les listes chaînées .....</b>	<b>141</b>
Les structures de données dynamiques .....	142
La bibliothèque de modèles Standard .....	143





Les listes chaînées . . . . .	144
Comprendre les listes chaînées . . . . .	144
Un programme gestionnaire simple . . . . .	148
La méthode HowMany() . . . . .	150
Insert() en détail . . . . .	152
Utiliser new . . . . .	158
new et delete . . . . .	158
Utiliser notre liste chaînée simple dans Decryptix! . . . . .	165
Exécuter Game . . . . .	172
Résoudre le problème avec une méthode membre . . . . .	173
Surcharger un opérateur . . . . .	174
Comment surcharger un opérateur . . . . .	174
Passer des objets par valeur . . . . .	175
Pourquoi une référence ? . . . . .	176
<b>7. Les méthodes canoniques . . . . .</b>	<b>179</b>
Surcharger une méthode . . . . .	179
Les classes Shape . . . . .	180
Surcharger les constructeurs . . . . .	180
Les méthodes d'office . . . . .	181
Le constructeur par défaut . . . . .	181
Quand utilisez-vous un constructeur fourni par le compilateur ? . . . . .	184
Le destructeur par défaut . . . . .	185
Le constructeur de copie . . . . .	186
Pourquoi le constructeur de copie par défaut n'est-il pas suffisant ? . . . . .	194
Ecrire votre propre constructeur de copie . . . . .	199
L'opérateur d'affectation . . . . .	200
Cela ressemble à une affectation, mais ce n'en est pas une . . . . .	207
De retour aux listes chaînées . . . . .	207
<b>8. Le polymorphisme . . . . .</b>	<b>209</b>
La spécialisation . . . . .	210
Les avantages de la spécialisation . . . . .	211
Le polymorphisme . . . . .	211
Les types de données abstraits . . . . .	212
Comment est-ce implémenté en C++ . . . . .	212
La syntaxe de l'héritage . . . . .	212
Remplacer des fonctions . . . . .	213
Les méthodes virtuelles . . . . .	213
Comment fonctionnent les fonctions virtuelles . . . . .	214
Les destructeurs virtuels . . . . .	216
Implémenter le polymorphisme . . . . .	219
Ajout d'une deuxième lettre . . . . .	232



Le fonctionnement d'operator[ ] .....	233
<b>9. Implémentation de modèles .....</b>	<b>235</b>
Créer l'historique .....	235
Les opérateurs surchargés .....	237
Envoyer des données de la classe à cout .....	238
Définition dans le fichier d'en-tête .....	245
Créer le type non paramétré d'abord .....	248
Instancier le modèle .....	248
L'objet History .....	250
Tirer parti de l'historique .....	257
La mise au point .....	258
<b>10. Amélioration de la bibliothèque de modèles Standard .....</b>	<b>259</b>
Convertir Decryptix! à la bibliothèque de modèles Standard .....	259
Les classes Collection .....	260
Travailler avec les vecteurs .....	260
Examiner le résultat .....	271
Initialiser le jeu .....	271
La méthode Play() .....	272
Afficher le contenu d'un vecteur .....	273
Etablir le score de la proposition .....	274
Créer l'historique .....	274
ShowHistory .....	275
Examiner les indices .....	276
<b>11. Les déductions de l'ordinateur .....</b>	<b>277</b>
Apprendre à l'ordinateur à deviner .....	277
Capturer les règles .....	278
Les classes Decrypter .....	279
Implémenter Game .....	282
Les boucles sans fin .....	287
Les propositions du joueur Human .....	288
Les déductions de l'ordinateur .....	297
Créer l'objet Computer .....	300
<b>12. Délégation de responsabilité .....</b>	<b>307</b>
Affecter une responsabilité .....	307
Comprendre le paramètre par défaut .....	316
SmartChar en détail .....	316
Comment ça marche .....	317
L'ordinateur joue .....	318
Créer un objet SmartString .....	318



Computer :Play() .....	321
Générer les propositions : généralités .....	322
Générer une proposition : détails .....	324
La logique de CanEliminateCharacters .....	325
CanEliminateCharacters en détail .....	325
GenerateAGuess .....	329
IsConsistent en détail .....	330
L'intérêt d'être statique .....	334
Reprendre l'analyse .....	334
Eliminer les caractères bien placés .....	336
<b>13. La persistance</b> .....	<b>339</b>
La persistance d'objet .....	339
Les types de persistance .....	339
Concevoir la persistance .....	340
Les classes Reader et Writer .....	341
L'interface utilisateur .....	344
L'écriture de Game .....	361
Ecrire Game étape par étape .....	361
L'écriture d'objets polymorphes .....	362
<b>14. Introduction des exceptions</b> .....	<b>381</b>
Lorsque quelque chose d'exceptionnel se produit .....	381
Dérouler la pile .....	388
Lire l'objet sur le disque .....	390
Restaurer un objet Human .....	398

## ANNEXES

<b>A. Nombres binaires et hexadécimaux</b> .....	<b>401</b>
Les autres bases .....	402
Le tour des bases .....	403
Le système binaire .....	404
Pourquoi la base deux ? .....	405
Bits, octets et "nybbles" .....	405
Qu'est-ce qu'un ko ? .....	405
Les nombres binaires .....	406
Le système hexadécimal .....	406
<b>B. Priorité des opérateurs</b> .....	<b>411</b>
<b>Index</b> .....	<b>413</b>