

---

# ***UN AUTRE REGARD SUR LE C++***

***Edition 2014***

**Jean Louis VINOLA**

1<sup>er</sup> édition – Septembre 2014 – Version 1.140912

Auteur VINOLA jean Louis (vinolajl@gmail.com)

Le code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5.2° et 3°a, d'une part, que les "copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective" et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, " toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite " (art.L.122-4). Cette représentation ou reproduction par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du code de la propriété intellectuelle.

Les informations contenues dans cet ouvrage sont données à titre indicatif et n'ont aucun caractère exhaustif voire certain. À titre d'exemple non limitatif, cet ouvrage peut présenter des spécificités et comportements des logiciels évoqués qui ne seront plus d'actualité ou valables au moment où vous en prendrez connaissance. Aussi ces informations ne sauraient engager la responsabilité de l'auteur. L'auteur ne pourra être tenu responsable de toute omission, erreur ou lacune qui aurait pu se glisser dans ce document ainsi que des conséquences, quelles qu'elles soient, qui résulteraient des informations et indications fournies ainsi que de leur utilisation.

Tous les produits et programmes cités dans cet ouvrage sont protégés, et les marques déposées par leurs titulaires de droits respectifs. Cet ouvrage n'est ni édité, ni produit par le(s) propriétaire(s) de(s) programme(s) sur le(s) quel(s) il porte. Les marques ne sont utilisées qu'à seule fin de désignation des produits en tant que noms de ces derniers.

Les informations contenues dans ce document pourront faire l'objet de modifications sans préavis.

ISBN : 978-2-9550525-0-1

Ce livre a été publié sur [www.bookelis.com](http://www.bookelis.com)

Impression réalisée par :  
BOOKELIS  
38 Parc du Golf,  
13 856 Aix-en-Provence.

---

*Note à l'intention du lecteur :*

*Tout au long de cet ouvrage, l'auteur tente d'apporter au lecteur un autre regard sur le C++.*

*Les notions fondamentales du langage sont certes présentées et exploitées à travers de nombreux exemples. Mais la véritable valeur ajoutée de cet ouvrage est qu'elles sont associées à diverses réflexions menant le lecteur à une compréhension des rouages internes du langage menant à la génération du code.*

*En présentant les mécanismes internes de l'analyseur syntaxique du langage C++, la façon avec laquelle le compilateur interprétera le code source, la transposition en langage C, cet ouvrage permet au lecteur de prendre conscience de la dimension « évolution » du C++ face au C. Cet ouvrage apporte un regard différent sur le codage à l'aide d'objets pouvant être transposé vers d'autres langages considérés comme non-objets. La compréhension des mécanismes internes de génération de code permet aussi au lecteur d'obtenir un code plus efficace, point particulièrement utile lorsqu'il est confronté à des plateformes équipées de processeurs et/ou de mémoires dont la capacité est limitée et dont les contraintes en termes de consommation ou de déperdition énergétique sont fortes.*

*Aujourd'hui, le codage à l'aide de langages objets est réellement à la portée de tout développeur. Mais l'expérience montre que la bonne connaissance des mécanismes internes d'un langage est un atout majeur pour l'obtention d'un code fiable, maintenable, évolutif, adapté aux contraintes de la plateforme cible et dont les compromis (vitesse de traitement, taille de l'exécutable et des données) sont choisis et non pas subis.*

*S'il est aisé de coder, il en est tout autre de bien coder et ce, en toute conscience des conséquences de nos choix. C'est ainsi que dans cet ouvrage, de nombreux exemples et démonstrations permettront de prouver l'action du compilateur offrant ainsi au lecteur la possibilité de prendre conscience de l'impact de ses choix sur l'activité processeur et sur l'occupation mémoire.*

*C'est pour tenter d'atteindre cet objectif, que l'auteur de ce document, fort d'une réelle expérience industrielle dans le domaine, vous fait partager son analyse du langage C++.*

*Tout au long de cet ouvrage, les spécificités du langage seront mises en œuvre grâce à de nombreux exemples pouvant être validées par le lecteur. De plus, un fil rouge est proposé à travers une problématique constante dans laquelle de nouvelles fonctionnalités apparaissent au fur et à mesure de la découverte de nouvelles notions. Le lecteur est donc invité à implémenter les solutions présentées.*

#### **ATTENTION :**

*Cet ouvrage traite des notions fondamentales de la programmation objet à l'aide du langage C++. La bonne connaissance du langage C et du développement séquentiel est nécessaire à la compréhension des éléments présentés ici. Nous invitons donc le lecteur à consulter les ouvrages relatifs à ces notions du même auteur avant la lecture de celui-ci.*

---

# Table des matières

<b>1</b>	<b><u>INSTALLATION DES OUTILS DE DÉVELOPPEMENT C++ .....</u></b>	<b><u>11</u></b>
1.1	OUTILS DE COMPILATION ET SYSTEMES .....	11
1.2	ENVIRONNEMENT NETBEANS .....	12
<b>2</b>	<b><u>LE C++ ET L'OBJET, RÉVOLUTION OU ÉVOLUTION .....</u></b>	<b><u>17</u></b>
2.1	NOTION « D'APPROCHE OBJET ».....	17
2.2	IMPLEMENTATION CLASSIQUE D'UN OBJET EN LANGAGE C .....	19
2.2.1	MODELE GENERAL EN LANGAGE C .....	22
2.3	LE BESOIN ACTUEL D'UNE EVOLUTION D'ECRITURE .....	22
2.3.1	BESOIN D'EVOLUTION VERS UNE SYNTAXE PLUS NATURELLE .....	23
2.3.2	BESOIN D'EVOLUTION VERS UNE SYNTAXE PLUS RESTRICTIVE .....	25
2.3.3	BESOIN D'EVOLUTION POUR UNE MEILLEURE SCALABILITE .....	27
2.4	LA REPONSE C++ .....	28
<b>3</b>	<b><u>C++ UN LANGAGE DE CLASSES .....</u></b>	<b><u>33</u></b>
3.1	CLASSES ET OBJETS.....	33
3.2	CLASSE A BASE DE « STRUCT ».....	34
3.3	ROLE DE THIS .....	37
3.3.1	ÉCRITURE DE THIS OPTIONNELLE .....	38
3.4	ÉCRITURE D'UNE CLASSE AU SEIN D'UN PROJET .....	39
3.5	CREATION ET UTILISATION D'INSTANCES .....	40
3.6	VALIDATION DU CODE GENERE D'UNE CLASSE :.....	46
3.6.1	TAILLE D'UNE CLASSE .....	46
3.6.2	ROLE DU POINTEUR THIS .....	47
3.6.3	PASSAGE EN PARAMETRE DE L'ADRESSE DE L'OBJET .....	48
3.7	DEVELOPPEMENT « INLINE » .....	50
3.8	METHODES « INLINE » .....	51
3.9	CONTROLE D'ACCES .....	53
3.10	NOTION D'ENCAPSULATION DE DONNEES ET DE COMPORTEMENT .....	56
3.11	CLASSE A BASE DE « CLASS ».....	59
3.12	ETAT INITIAL D'UN OBJET .....	60
3.13	ASSOUPLISSEMENT DU CONTROLE D'ACCES : LA PROCURATION « FRIEND ».....	61
<b>4</b>	<b><u>ÉLÉMENTS SYNTAXIQUES SUPPLÉMENTAIRES.....</u></b>	<b><u>65</u></b>
4.1	NOUVEAUX TYPES .....	65
4.2	SYMBOLES DE SUBSTITUTIONS .....	66
4.3	LES ESPACES DE NOM.....	67
4.4	OBJETS DE CANAUX STANDARDS STDIN, STDOUT, STDERR.....	71
4.4.1	L'OBJET COUT .....	71
4.4.2	L'OBJET CIN .....	74
4.4.3	L'OBJET CERR .....	75
4.4.4	L'OBJET CLOG .....	75
4.5	ADRESSAGE PAR REFERENCE : LE SYMBOLE & .....	76
4.5.1	PASSAGE DE PARAMETRES PAR REFERENCE : UN VRAI PASSAGE PAR ADRESSE ! .....	78
4.5.2	RETOURNER UNE REFERENCE .....	80

<b>4.6</b>	<b>PROTECTION DES DONNEES CONSTANTES.....</b>	<b>81</b>
4.6.1	INTERDICTION D'ECRITURE D'UNE VARIABLE.....	82
4.6.2	INTERDICTION D'ECRITURE D'UNE ZONE MEMOIRE POINTEE.....	83
4.6.3	PROTECTION CUMULEE.....	85
4.6.4	ECRITURE ALTERNATIVE.....	85
4.6.5	PROTECTION CONTRE L'ECRITURE DES REFERENCES.....	86
4.6.6	EXPLOITATION DANS UNE CLASSE.....	87
4.6.7	PROTECTION CONTRE L'ECRITURE INDIRECTE.....	89
4.6.8	ÉLÉMENTS MUTABLES.....	90
<b>4.7</b>	<b>POLYMORPHISME AD HOC OU SURCHARGE DE METHODE .....</b>	<b>92</b>
<b>4.8</b>	<b>VALEUR PAR DEFAULT.....</b>	<b>94</b>
<b>4.9</b>	<b>CONSTANTE NULLPTR.....</b>	<b>96</b>
<b>4.10</b>	<b>CONSTANTE EVALUEE EN PRE-COMPILATION .....</b>	<b>96</b>
<b>4.11</b>	<b>UNIFORMITE DE L'INITIALISATION .....</b>	<b>97</b>
<b>4.12</b>	<b>GENERATION D'INFORMATIONS NOMINATIVES DE DEBOGAGE.....</b>	<b>98</b>
<b>4.13</b>	<b>IDENTIFICATEUR DE TYPES .....</b>	<b>99</b>
<b>5</b>	<b><u>CYCLE DE VIE D'UN OBJET .....</u></b>	<b><u>103</u></b>
<b>5.1</b>	<b>CONSTRUCTION D'UN OBJET.....</b>	<b>103</b>
5.1.1	LIEU DE LA CONSTRUCTION.....	105
5.1.2	SURCHARGE DU CONSTRUCTEUR PAR DEFAULT.....	106
5.1.3	CONTROLE DE LA PHASE DE POST-CREATION DE L'OBJET.....	108
5.1.4	SURCHARGE MULTIPLE DU CONSTRUCTEUR .....	110
5.1.5	DELEGATION DE CONSTRUCTION .....	112
5.1.6	CONSTRUCTION D'OBJET ANONYME .....	113
5.1.7	CONSTRUCTEUR DE CONVERSION .....	114
5.1.8	CONSTRUCTION EXPLICITE.....	115
5.1.9	SURCHARGE DU CONSTRUCTEUR DE COPIE.....	117
5.1.10	CONSTRUCTEUR DE COPIE EXPLICITE.....	120
5.1.11	SUPPRESSION DU CONSTRUCTEUR DE COPIE PAR DEFAULT.....	121
<b>5.2</b>	<b>DESTRUCTION D'UN OBJET.....</b>	<b>122</b>
5.2.1	SURCHARGE DU DESTRUCTEUR.....	123
<b>6</b>	<b><u>GESTION DYNAMIQUE DES OBJETS .....</u></b>	<b><u>127</u></b>
<b>6.1</b>	<b>CREATION DYNAMIQUE : NEW .....</b>	<b>127</b>
6.1.1	CREATION DYNAMIQUE D'INSTANCES PRIMITIVES .....	130
6.1.2	CREATION DYNAMIQUE D'OBJETS.....	131
6.1.3	CREATION DYNAMIQUE D'UN ENSEMBLE D'OBJETS.....	133
6.1.4	DESACTIVATION DE L'ALLOCATION MEMOIRE : NEW DE PLACEMENT.....	135
6.1.5	GESTION D'ERREUR D'ALLOCATION .....	137
<b>6.2</b>	<b>DESTRUCTION DYNAMIQUE : DELETE .....</b>	<b>138</b>
<b>7</b>	<b><u>LES OPÉRATEURS .....</u></b>	<b><u>143</u></b>
<b>7.1</b>	<b>OPERATEURS UNAIRES .....</b>	<b>145</b>
<b>7.2</b>	<b>OPERATEURS BINAIRES .....</b>	<b>147</b>
7.2.1	SURCHARGE DE L'OPERATEUR D'AFFECTATION (COPIE) PAR DEFAULT.....	150
7.2.2	SUPPRESSION DE L'OPERATEUR D'AFFECTATION PAR DEFAULT.....	151
<b>7.3</b>	<b>EXPLOITATION DES OPERATEURS BINAIRES AVEC LES OBJETS FLUX .....</b>	<b>152</b>
7.3.1	FONCTION OPERATEUR << AVEC UN OBJET FLUX SORTANT.....	153
7.3.2	FONCTION OPERATEUR >> AVEC UN OBJET FLUX ENTRANT .....	155
<b>7.4</b>	<b>OPERATEUR DE TRANSTYPAGE .....</b>	<b>157</b>
<b>7.5</b>	<b>OPERATEURS LITTERAUX UTILISATEURS.....</b>	<b>160</b>

<b>7.6</b>	<b>OPERATEUR D'ALLOCATION ET DE LIBERATION .....</b>	<b>161</b>
7.6.1	MECANISME D'ALLOCATION : L'OPERATEUR NEW.....	161
7.6.2	MECANISME DE LIBERATION : L'OPERATEUR DELETE.....	166
<b>7.7</b>	<b>LES FONCTEURS .....</b>	<b>169</b>
<b>8</b>	<b><u>SÉMANTIQUE MOVE : LE DEPLACEMENT D'OBJET .....</u></b>	<b>175</b>
<b>8.1</b>	<b>NOTION DE LVALUE ET RVALUE.....</b>	<b>177</b>
<b>8.2</b>	<b>LES REFERENCES LVALUE ET RVALUE .....</b>	<b>178</b>
<b>8.3</b>	<b>OPERATEUR DE DEPLACEMENT (MOVE).....</b>	<b>182</b>
<b>8.4</b>	<b>CONSTRUCTEUR DE DEPLACEMENT (MOVE) .....</b>	<b>184</b>
<b>9</b>	<b><u>COMPOSITION AVANCÉE D'UNE CLASSE.....</u></b>	<b>191</b>
<b>9.1</b>	<b>DONNEE MEMBRE LVALUE REFERENCE.....</b>	<b>191</b>
<b>9.2</b>	<b>DONNEE MEMBRE STATIQUE : LE PARTAGE DE ZONE MEMOIRE.....</b>	<b>194</b>
<b>9.3</b>	<b>FONCTION MEMBRE STATIQUE .....</b>	<b>199</b>
<b>9.4</b>	<b>TYPE ET CLASSES « NESTED » (IMBRIQUEES) .....</b>	<b>202</b>
<b>10</b>	<b><u>ÉVOLUTION DE LA PRECOMPILATION : GÉNÉRICITÉ ET MÉTA-PROGRAMMATION .....</u></b>	<b>207</b>
<b>10.1</b>	<b>LA META-PROGRAMMATION EN C .....</b>	<b>208</b>
<b>10.2</b>	<b>ÉVOLUTION DE SYNTAXE : LES TEMPLATES.....</b>	<b>212</b>
10.2.1	GENERICITE DE TYPE .....	213
10.2.2	GENERICITE DE CONSTANCE .....	219
10.2.3	GENERICITE TYPES ET CONSTANTES.....	222
10.2.4	PARAMETRES GENERIQUES PAR DEFAUT .....	223
10.2.5	RENOMMAGE D'UN TYPE GENERIQUE .....	223
<b>11</b>	<b><u>LES CLASSES CONTENEURS STL .....</u></b>	<b>227</b>
<b>11.1</b>	<b>LES CLASSES CONTENEURS DE SEQUENCES.....</b>	<b>228</b>
11.1.1	LA CLASSE STRING <STRING>.....	228
11.1.2	LES ARRAYS <ARRAY> .....	231
11.1.3	LES VECTORS <VECTOR> .....	233
11.1.4	LES DEQUES <DEQUE> .....	236
<b>11.2</b>	<b>LES CLASSES CONTENEURS DE SEQUENCE A ITERATEURS .....</b>	<b>238</b>
11.2.1	MODELE GENERAL CONTENEUR – ITERATEUR .....	238
11.2.2	LES FORWARD LISTS <FORWARD_LIST> .....	242
11.2.3	LES LISTS <LIST>.....	244
<b>11.3</b>	<b>LES CLASSES CONTENEURS ASSOCIATIFS .....</b>	<b>246</b>
11.3.1	LES MAPS <MAP> .....	246
11.3.2	LES MUTIMAPS <MAP> .....	248
11.3.3	LES SETS <SET>.....	250
11.3.4	LES MULTISSETS <SET> .....	251
<b>11.4</b>	<b>CLASSES CONTENEURS ADAPTATEURS .....</b>	<b>252</b>
11.4.1	STACK <STACK> .....	252
11.4.2	QUEUE <QUEUE> .....	253
11.4.3	PRIORITY_QUEUE <QUEUE>.....	254
11.4.4	LES BITSET <BITSET> .....	255
<b>11.5</b>	<b>RESUME DES COMPLEXITES .....</b>	<b>256</b>
<b>11.6</b>	<b>LES TRAITEMENTS ALGORITHMIQUES &lt;ALGORITHM&gt; .....</b>	<b>256</b>

<b>12</b>	<b>RELATION ET ASSOCIATION DE CLASSES .....</b>	<b>261</b>
12.1	LA RELATION « UTILISATION » (DEPENDANCE).....	261
12.2	LA RELATION « A CONNAISSANCE DE ».....	265
12.3	LA RELATION « EST CONSTITUE DE » : L'AGREGATION .....	268
12.4	LA RELATION « EST COMPOSE DE » : LA COMPOSITION .....	271
12.4.1	AGREGATION FORTE : UNE ALTERNATIVE A LA COMPOSITION .....	276
12.5	NOTION DE MULTIPLICITE .....	277
12.5.1	CAS D'UN ENSEMBLE FINI.....	278
12.5.2	CAS D'UN ENSEMBLE NON FINI.....	279
<b>13</b>	<b>ÉVOLUTION ALGORITHMIQUE.....</b>	<b>285</b>
13.1	EXCEPTION .....	285
13.1.1	NOTION D'EXCEPTION LOGICIELLE .....	286
13.1.2	DEFINITION DE LA PORTEE DE LA GESTION D'EXCEPTION : LE BLOC TRY .....	287
13.1.3	GESTIONNAIRE DES EXCEPTIONS : LES FONCTIONS CATCH .....	288
13.1.4	GESTIONNAIRE D'EXCEPTIONS PAR DEFAUT .....	290
13.1.5	PROPAGATION DES EXCEPTIONS.....	291
13.1.6	TRAITEMENT DES EXCEPTIONS NON GEREES .....	292
13.1.7	RESTRICTION DE PROPAGATION .....	293
13.1.8	LES CLASSES D'EXCEPTION STD .....	295
13.1.9	LES CLASSES D'EXCEPTION SPECIALISEES .....	297
13.2	INFERENCE DE TYPE .....	298
13.2.1	L'INFERENCE VIA « AUTO ».....	298
13.2.2	L'INFERENCE VIA « DECLTYPE ».....	300
13.2.3	L'INFERENCE DANS LE PROTOTYPE D'UNE FONCTION « ➔ » .....	302
13.3	STRUCTURE FOR .....	304
13.4	LES FONCTIONS LAMBDA .....	305
13.4.1	DECLARATION ET INVOCATION D'UNE FONCTION LAMBDA.....	305
13.4.2	EXPLOITATION D'UNE LAMBDA EN PARAMETRE.....	307
13.4.3	EXPLOITATION DE LA CAPTURE.....	309
13.4.4	EXPLOITATION EN LIEU ET PLACE DE FONCTEURS .....	310
<b>14</b>	<b>ÉLÉMENTS AGRÉGÉS : PROBLÉMATIQUE DE SÉRIALISATION.....</b>	<b>315</b>
14.1	ENCODAGE LV ET TLV .....	316
<b>15</b>	<b>SPÉCIALISATION ET HÉRITAGE .....</b>	<b>325</b>
15.1	HERITAGE SIMPLE .....	326
15.1.1	HERITAGE PUBLIC.....	327
15.1.2	HERITAGE PROTECTED .....	330
15.1.3	HERITAGE PRIVATE .....	331
15.1.4	CONTROLE D'ACCES SPECIFIQUE (USING) .....	332
15.1.5	L'HERITAGE EN TANT QUE TECHNIQUE DE RENOMMAGE.....	334
15.2	L'HERITAGE, UNE AUTRE IMPLEMENTATION DE LA COMPOSITION .....	335
15.3	CONSTRUCTEUR ET DESTRUCTION D'OBJETS HERITES.....	337
15.4	SURCHARGE DE METHODES HERITEES : UN AUTRE POLYMORPHISME AD HOC .....	341
15.5	APPEL DE METHODE PLEINEMENT QUALIFIE.....	341
15.6	DETERMINATION DYNAMIQUE DES ADRESSES DES METHODES : LA TMV.....	344
15.7	CONTRAINTES DE CONSTRUCTION .....	352
15.8	METHODE VIRTUELLE PURE ET CLASSES ABSTRAITES .....	354
15.9	POINTEUR DE FAMILLE : UN AUTRE POLYMORPHISME DYNAMIQUE .....	357

---

<b>15.10</b>	<b>DESTRUCTION DE POINTEUR DE FAMILLE.....</b>	<b>360</b>
<b>15.11</b>	<b>POLYMORPHISME STATIQUE ANTICIPE .....</b>	<b>361</b>
<b>15.12</b>	<b>HERITAGE MULTIPLE.....</b>	<b>364</b>
<b>15.13</b>	<b>PROBLEMATIQUES D'AMBIGUÏTE .....</b>	<b>365</b>
15.13.1	LEVEE D'AMBIGUÏTE SUR LES DONNEES MEMBRES .....	365
15.13.2	LEVEE D'AMBIGUÏTE SUR LES METHODES .....	366
<b>15.14</b>	<b>PROBLEMATIQUE DE MAILLE OU DE DIAMANT : HERITAGE VIRTUEL .....</b>	<b>367</b>
<b>15.15</b>	<b>REPARTITION DES VPOINTERS .....</b>	<b>370</b>
<b>15.16</b>	<b>EXPLOITATION DE L'HERITAGE MULTIPLE.....</b>	<b>370</b>
<b>16</b>	<b><u>GESTION EVOLUEE DES POINTEURS .....</u></b>	<b><u>373</u></b>
<b>16.1</b>	<b>LA CLASSE UNIQUE_PTR&lt;&gt;.....</b>	<b>373</b>
<b>16.2</b>	<b>LA CLASSE SHARED_PTR&lt;&gt; .....</b>	<b>375</b>
<b>17</b>	<b><u>NOTE DE L'AUTEUR .....</u></b>	<b><u>379</u></b>

---

# ***INSTALLATION DES OUTILS DE DÉVELOPPEMENT C++***

<b>1</b>	<b><u>INSTALLATION DES OUTILS DE DEVELOPPEMENT C++ .....</u></b>	<b><u>11</u></b>
<b>1.1</b>	<b>OUTILS DE COMPILATION ET SYSTEMES .....</b>	<b>11</b>
<b>1.2</b>	<b>ENVIRONNEMENT NETBEANS .....</b>	<b>12</b>



# 1 INSTALLATION DES OUTILS DE DÉVELOPPEMENT C++

Pour l'ensemble de cet ouvrage, l'environnement de développement choisi est composé de l'IDE (Environnement de Développement Intégré) « Netbeans », des outils de compilation GNU GCC et des commandes en ligne compatibles Linux (Bourne Shell) sous Windows. Tous ces éléments sont disponibles gratuitement. Il sera bien évidemment possible de mettre en œuvre les différentes techniques présentées dans cet ouvrage à l'aide de tout autre EDI. Ce chapitre a pour objectif d'accompagner le lecteur à l'installation et la configuration de l'environnement de développement choisi par l'auteur.

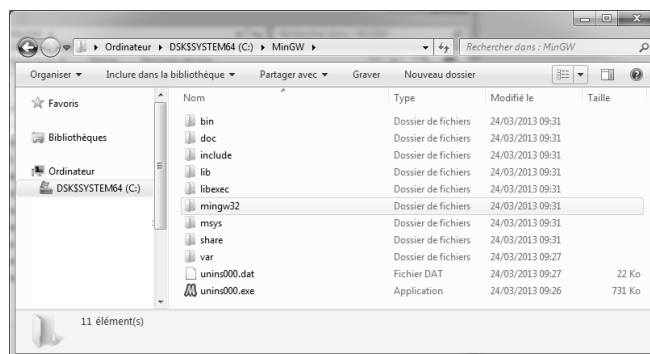
## 1.1 Outils de compilation et systèmes

Les outils GCC sont disponibles via un paquetage dénommé MinGW (Minimalist GNU for Windows). Les applications développées à l'aide de ces outils dépendront d'un certain nombre de DLL fournies par MinGW ainsi que d'autres présentes en standard dans le système d'exploitation. Le déploiement d'une application développée à l'aide des outils MinGW sera de ce fait aisé.

Comme toujours, la génération de chaque application s'effectuera en analysant un fichier Makefile. Pour rappel, ce fichier contient la liste des fichiers sources et des bibliothèques nécessaires, les commandes de compilation et d'édition des liens ainsi que celles de création de divers répertoires indispensables à l'obtention du fichier exécutable. La commande Make analysera le fichier Makefile du projet et s'appuiera donc sur des commandes Shell de Linux pour atteindre l'objectif fixé. Le fait que nous soyons sous windows, nécessite donc l'installation de ces outils compatibles Linux sur notre plateforme.

Les outils de ligne de commandes compatibles Linux nécessaires à la commande Make sont rassemblés dans un paquetage nommé MSYS (SYStem Minimal).

L'ensemble de ces outils sera installé à la racine de notre système dans un répertoire MinGW :

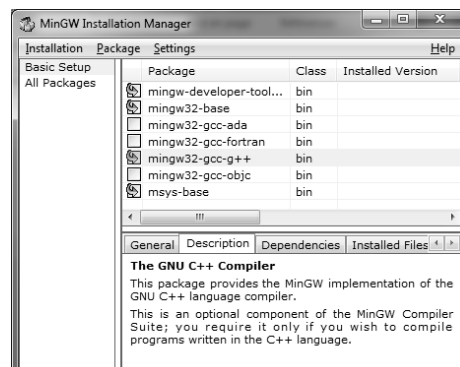


L'installation peut être réalisée de deux façons : Soit copier le répertoire MinGW venant d'une installation déjà réalisée sur une autre machine (~400 Mo), soit, utiliser un outil permettant de télécharger la dernière version de la distribution.

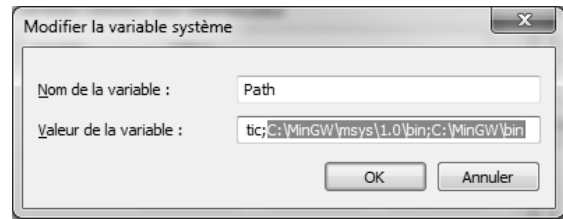
Dans ce dernier cas, le lien « *download* » de la page <http://mingw.org/> pointerait vers un installateur dont le nom sera de la forme : [mingw-get-setup.exe](#)

L'installation permettra de préciser les compilateurs souhaités ainsi que les outils complémentaires à installer :

Le choix se portera vers MinGW-developer-toolkit (qui contient MSys), MinGW32-base (qui contient le C) et MinGW32-gcc-g++ (qui contient le C++).



Quelle que soit la méthode d'installation, il faudra impérativement penser à mettre à jour la variable d'environnement PATH afin de permettre au système d'exploitation de parcourir les répertoires « bin » de Mingw et de MSys lors de la recherche des outils à exécuter (accessible via les « propriétés » du poste de travail en cliquant sur « propriétés du système avancé »):



*Remarque : ne pas poursuivre l'installation si cette dernière étape n'a pas été réalisée !*

La version Minimale du compilateur dont nous avons besoin dans cet ouvrage est la 4.7.2.

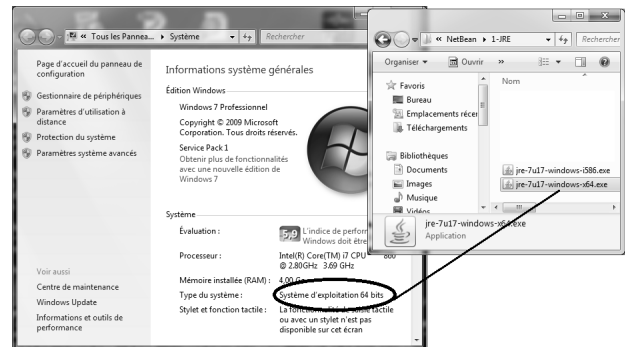
Remarque : La commande « `gcc -v` » nous indique la version installée :

```
C:\Users\Administrateur>gcc -v
Utilisation des specs internes.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=c:/mingw/bin/./libexec/gcc/mingw32/4.7.2/lto-wrapper.exe
Target: mingw32
Configuré avec: ../gcc-4.7.2/configure --enable-languages=c,c++,ada,fortran,objc,obj-c++ --disable-sjlj-exceptions --with-dwarf2 --enable-shared --enable-libgomp --disable-win32-registry --enable-libstdcxx-debug --disable-build-poststage1-with-cxx --enable-version-specific-runtime-libs --build=mingw32 --prefix=/mingw
Modèle de thread: win32
gcc version 4.7.2 (GCC)
```

Pour information, la désinstallation de ces outils consiste simplement à supprimer le répertoire et mettre à jour le PATH.

## 1.2 Environnement NetBeans

L'IDE Netbeans a été développé en java. De ce fait, il est nécessaire d'installer une machine virtuelle JRE. Toute version supérieure à 7.u10 de JRE est compatible avec Netbeans. Il est toutefois extrêmement important d'installer une version en respectant la plateforme de votre OS (JRE 64 bits pour système 64 bits). Il est aussi recommandé d'installer la version disponible au moment du téléchargement de Netbeans.



La JRE est disponible au téléchargement via : <http://www.java.com/fr/download/manual.jsp>

Attention, en fonction du navigateur internet que nous utiliserons, le site de téléchargement peut nous indiquer une mauvaise version de java. Choisissons celle qui respecte la plateforme de l'OS (32 ou 64 bits) indépendamment de la version du navigateur.

Remarque : La commande `java -version` nous indique la version installée :

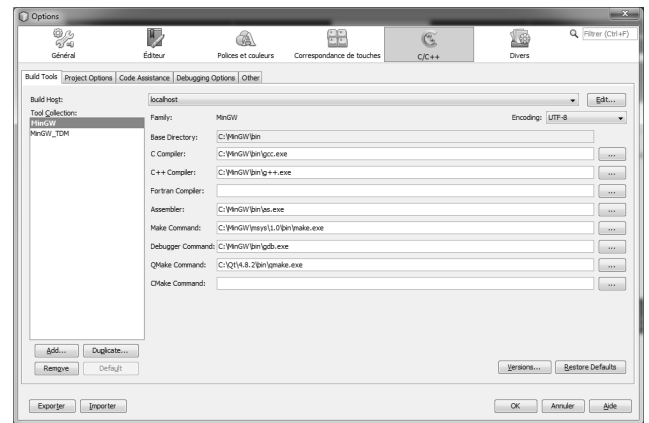
```
C:\Users\Administrateur>java -version
java version "1.7.0_17"
Java(TM) SE Runtime Environment (build 1.7.0_17-b02)
Java HotSpot(TM) 64-Bit Server VM (build 23.7-b01, mixed mode)
```

Netbeans est disponible au téléchargement via le lien : <http://netbeans.org/downloads/> Le choix se portera vers la version dénommée « C/C++ » qui est particulièrement adaptée (auto complétion du code, colorisation, Callgraph, ....) pour du développement en C et C++.

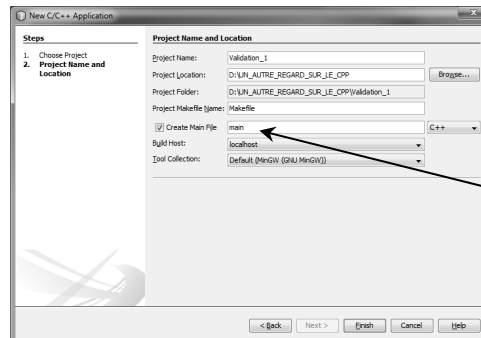
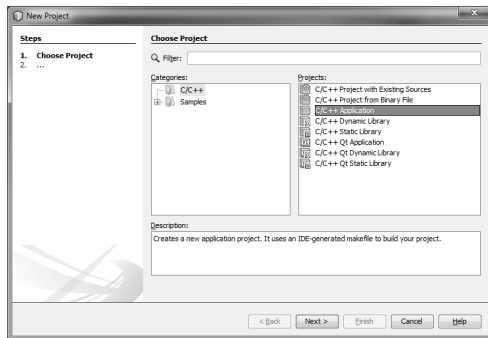
Pour la suite de cet ouvrage, la version 7.3 sera utilisée mais toute version supérieure sera tout aussi utilisable. Les options d'installation par défaut sont parfaitement adaptées à ce dont nous avons besoin.

À l'issue de l'installation, une vérification de la connaissance qu'aura Netbeans des outils de développement est à réaliser. Dans le menu *Outils/Options*, l'onglet *C/C++* sont affichés les « Build Tools » connus par le système :

Si la variable d'environnement PATH a été correctement alimentée, le « Tool collection » MinGw doit être présent.

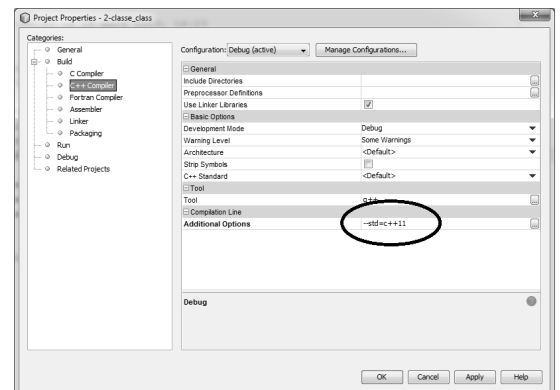


Netbeans est alors opérationnel pour la suite du document. La création des projets C++ s'effectuera via le menu « File/New Project ». Le choix se portera vers « C/C++ Application »

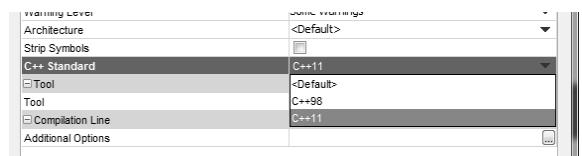


En profiter pour changer le nom par défaut (main) du fichier contenant le programme principal. Si plusieurs projets sont ouverts simultanément, il est alors judicieux de nommer le fichier à l'aide du nom du projet

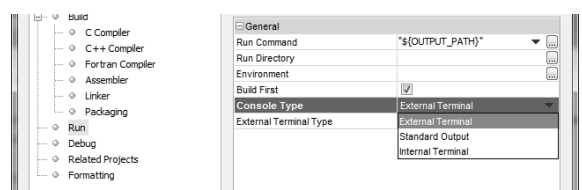
Dès la création d'un projet, il sera nécessaire de spécifier une option de compilation ayant pour objectif de préciser le fait que nous souhaitons utiliser la syntaxe de la version 11 du langage C++ (11, signifiant que le langage répond au formalisme décrit par les spécificités définies en 2011). Par défaut la syntaxe est opérationnelle mais à chaque utilisation d'une des particularités de celle-ci, la compilation nous informe de la chose sous forme de warning afin de rappeler que le code sera incompatible avec de très anciennes versions de compilateur. La mention « **-std=c++11** » sera ajoutée à « *Additional Options* » dans la catégorie « C++ Compiler »



Remarque : dans les dernières versions de NetBeans, il est tout aussi possible de forcer cette option de compilation en spécifiant le standard utilisé via la liste déroulante « *C++ Standard* » :



Les résultats de l'exécution de nos programmes s'effectuant par défaut dans une console interne à NetBeans, il sera nécessaire de préciser l'utilisation du shell de notre système (cmd sous windows) via l'option « *console type* » dans la catégorie « *Run* ». Le choix se portera vers « *External terminal* ».





---

# ***LE C++ ET L'OBJET, RÉVOLUTION OU ÉVOLUTION***

<b>2</b>	<b>LE C++ ET L'OBJET, REVOLUTION OU EVOLUTION .....</b>	<b>17</b>
<b>2.1</b>	<b>NOTION « D'APPROCHE OBJET ».....</b>	<b>17</b>
<b>2.2</b>	<b>IMPLEMENTATION CLASSIQUE D'UN OBJET EN LANGAGE C .....</b>	<b>19</b>
2.2.1	MODELE GENERAL EN LANGAGE C .....	22
<b>2.3</b>	<b>LE BESOIN ACTUEL D'UNE EVOLUTION D'ECRITURE .....</b>	<b>22</b>
2.3.1	BESOIN D'EVOLUTION VERS UNE SYNTAXE PLUS NATURELLE.....	23
2.3.2	BESOIN D'EVOLUTION VERS UNE SYNTAXE PLUS RESTRICTIVE .....	25
2.3.3	BESOIN D'EVOLUTION POUR UNE MEILLEURE SCALABILITE .....	27
<b>2.4</b>	<b>LA REPONSE C++ .....</b>	<b>28</b>