

# **Programmation avancée en C++**

## **GIF-1003**

### **Module 1 : Codage des algorithmes**

#### **1.1 Introduction**

T. Eude

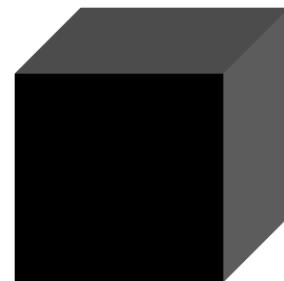


UNIVERSITÉ  
**LAVAL**

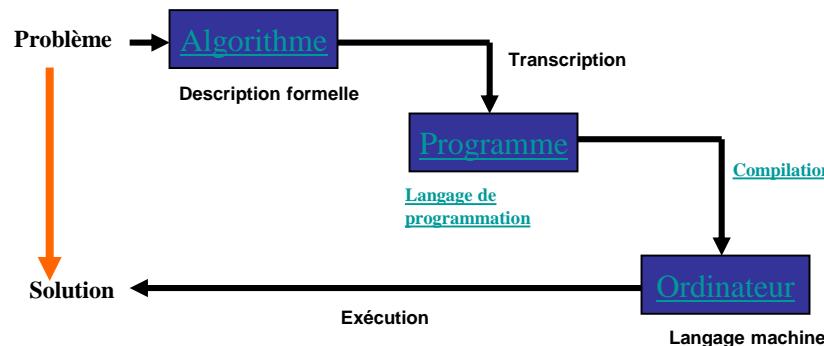
Département d'informatique  
et de génie logiciel

# Qu'est-ce qu'un programme?

Données en entrée



Données (résultat)  
en sortie



# Algorithme, programme et langage machine

Problème →

Algorithme

formelle

fonction [Dist, Pred] = DIJKSTRA(G, s)

Initialisations :

n = nombre de sommets du graphe G

Pred = tableau des prédécesseurs initialisé à 0

Dist = tableau des distances initialisé à  $+\infty$

Dist(s) = 0

W = matrice de poids

C = {1, 2, ..., n}

Traitements :

tant que  $C \neq \emptyset$

x = s

retire x de C

pour tout y dans C - {x}

si  $w_{xy} < Dist(y)$

alors

Dist(y) =  $Dist(x) + w_{xy}$

Pred(y) = x

fin si

fin pour

fin tant que

fin fonction

pas de r<sub>y</sub>)

r<sub>y</sub>)

Pred(y)

dist(y)

dist(x)

dist(r<sub>x</sub>)

dist(r<sub>y</sub>)

dist(r<sub>x</sub>)

# Les langages de programmation: Vocabulaire, syntaxe et sémantique

---

**Vocabulaire:** ensemble des symboles/termes utilisables pour construire des expressions

**Syntaxe:** ensemble des règles définissant la bonne construction des expressions

**Sémantique:** ensemble des règles permettant d'associer un sens à une expression (signification d'une expression)

# Les langages de programmation

[Language Popularity Index](#) - Web queries done on: 2013/07/01 18:49

Language category:

[any \\*](#)

123 entries.

Rank	Name	Share	Last month's share	Last year's share
1	C	17.668%	15.868%	16.825%
2	Java	14.720%	15.450%	20.381%
3	Objective-C	8.230%	8.516%	9.221%
4	C++	6.770%	7.544%	7.912%
5	Basic	5.457%	5.955%	7.592%
6	PHP	4.401%	4.144%	4.247%
7	Python	3.658%	3.363%	3.616%
8	C#	3.269%	3.444%	4.598%
9	Perl	2.566%	2.455%	2.459%
10	Ruby	1.918%	1.392%	1.576%

Language category:

[general-purpose \\*](#)

48 entries.

Rank	Name	Share
1	C	26.623%
2	Java	22.182%
3	Objective-C	12.401%
4	C++	10.202%
5	Basic	8.223%
6	C#	4.926%
7	Pascal	1.906%
8	Ada	1.688%
9	Fortran	1.201%
10	Forth	1.148%
11	D	0.955%

Language category:

[script \\*](#)

49 entries.

Rank	Name	Share
1	PHP	19.801%
2	Python	16.458%
3	Perl	11.545%
4	Ruby	8.628%
5	JavaScript	5.399%
6	R	4.898%
7	Lisp/Scheme	3.960%
8	MATLAB	3.488%
9	Bourne shell	2.757%
10	APL	2.316%
11	NXT-G	1.871%

Language category:

[other \\*](#)

26 entries.

Rank	Name	Share
1	Logo	14.778%
2	COBOL	14.597%
3	Prolog	10.099%
4	PL/SQL	8.939%
5	SAS	8.619%
6	LabView	8.317%
7	ABAP	6.482%
8	Transact-SQL	2.837%
9	Focus	2.728%
10	XSLT	2.713%
11	YACC	2.550%

The Language Popularity Index tool is a fully automatic, transparent, open-source and free tool to measure the popularity of programming languages on the Internet.

To see the bigger picture, please find below the positions of the top 10 programming languages of many years back. Please note that these are *average* positions for a period of 12 months.

Programming Language	2016	2011	2006	2001	1996	1991	1986
Java	1	1	1	3	17	-	-
C	2	2	2	1	1	1	1
C++	3	3	3	2	2	2	5
C#	4	5	6	11	-	-	-
Python	5	6	7	25	23	-	-
PHP	6	4	4	8	-	-	-
JavaScript	7	9	8	7	21	-	-
Visual Basic .NET	8	29	-	-	-	-	-
Perl	9	8	5	4	3	-	-
Ruby	10	10	21	32	-	-	-
Ada	27	16	16	17	7	4	2
Lisp	28	12	12	14	6	7	3
Pascal	62	13	17	15	4	3	7

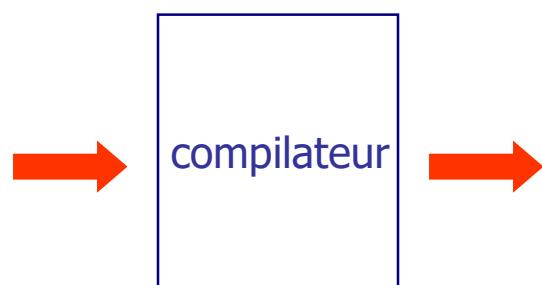
© 2016 TIOBE software BV (TIOBE checks more than 400 million lines of software code for its customers world-wide, realtime, each day. )

# Les compilateurs

Qu'est-ce qu'un compilateur ? (interpréteur?)

```
int main(void)
{
    int a;
    int b;
    int c;

    c = a + b;
    return 0;
}
```



```
00100010010101  
01100100011101  
11011101000110  
00110111011000  
01110011010100  
10110111101011  
11101101000010  
11101011000101  
01110101000100
```

Fichier source

Correspondance en langage machine

# Les compilateurs

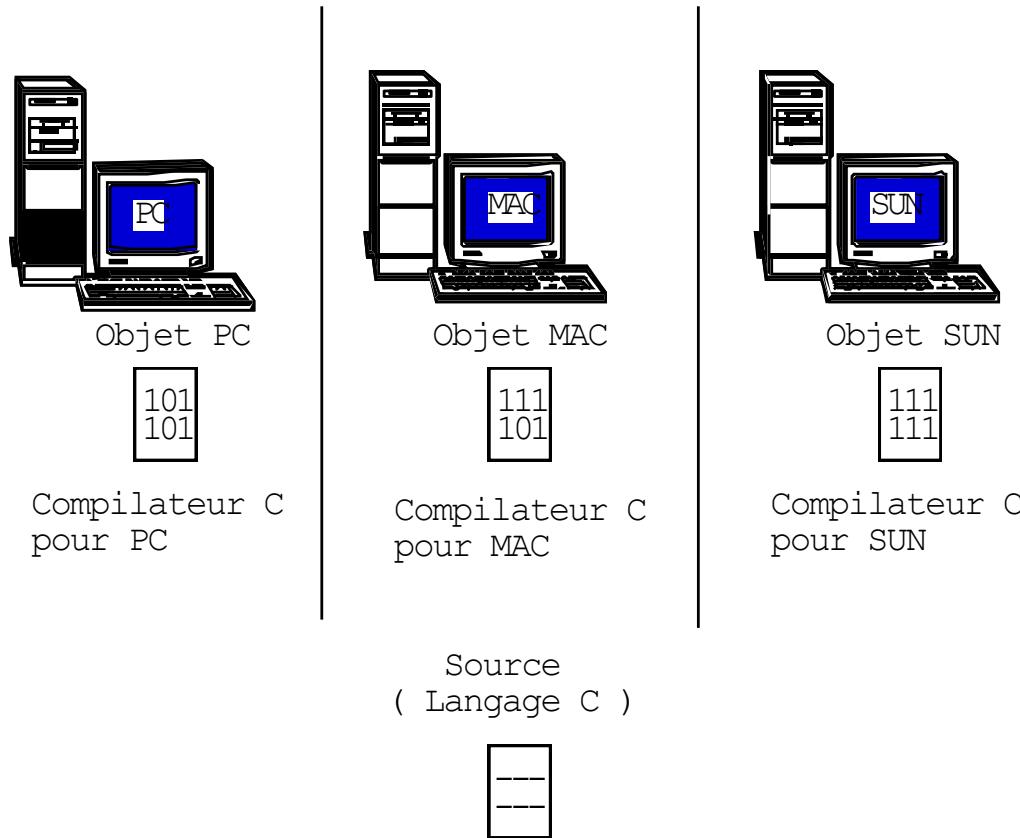
---

- Un compilateur donné est toujours spécifique au processeur pour lequel il a été conçu.
- Si l'on suit bien les standards établis pour les langages de programmation que l'on utilise, les compilateurs de plusieurs processeurs différents seront en mesure de comprendre nos programmes et de générer le code machine correspondant.



# Les compilateurs

## La portabilité d 'un programme



# Les compilateurs

## La portabilité d'un programme

- Certains des problèmes liés à la portabilité disparaissent avec une standardisation du langage : **ANSI** et **ISO Organization**.
- ⚠ Des fabricants du compilateur ajoutent des “extensions” au langage qui le rendent incompatible avec certains environnements de programmation.

# Langage de programmation : le C++

- est un langage de programmation permettant la programmation sous de multiples paradigmes comme
  - la programmation procédurale,
  - la programmation orientée objet
  - la programmation générique.
- Un des langages le plus utilisé au monde.
- n'appartient à personne
  - ➔ n'importe qui peut l'utiliser sans besoin d'une autorisation ou obligation de payer pour avoir le droit d'utilisation.



# Compilateurs C++

---

- **GNU Compiler Collection (UNIX, Windows, DOS, etc.)**
- Microsoft Visual C++ (Windows)
- Borland C++ Builder (Windows)
- Intel C++ Compiler (Windows, Linux, MacOS)
- Comeau C++ Compiler

# Environnements de développement

- Anjuta DevStudio
- Code::Blocks
- C++ Builder
- Dev-C++ et son probable successeur WxDev-C++
- **Eclipse avec le plugin CDT** 
- KDevelop
- Microsoft Visual C++
- NetBeans
- QtCreator
- Sun Studio
- Xcode

Environnement  
Officiel du cours



# **Programmation avancée en C++**

## **GIF-1003**

### **Module 1 : Codage des algorithmes**

1.2 Environnement de développement et processus de compilation

T. Eude



Département d'informatique  
et de génie logiciel

# Environnement de développement intégré (IDE)



Compilateur : g++

Éditeur de lien : g++

Débogueur : gdb

Documentation : doxygen, graphviz

Interface GUI : Framework Qt

...



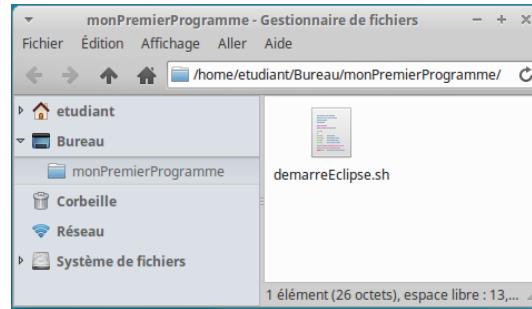
## Mon premier programme

---

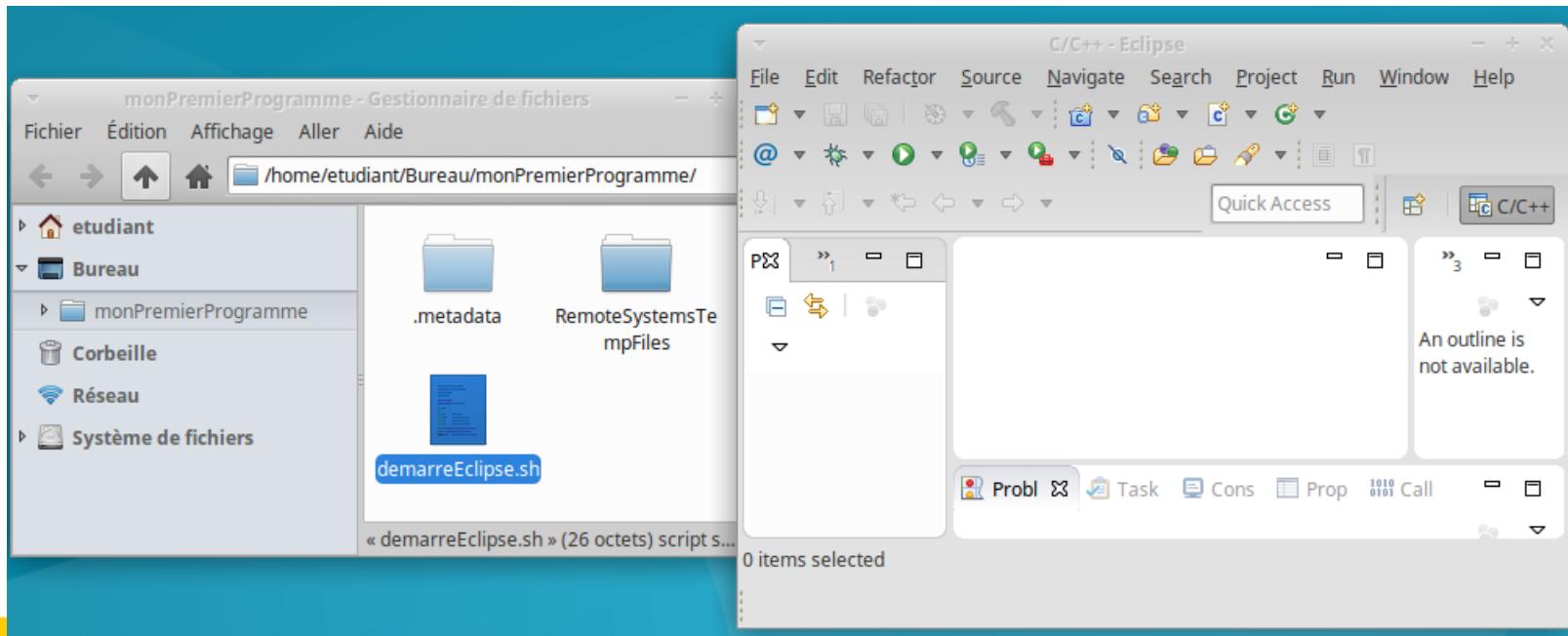
- Espace de travail (workspace)
- Projet
- Programme exécutable
- La compilation
- L'édition de lien
- L'exécution



# Espace de travail

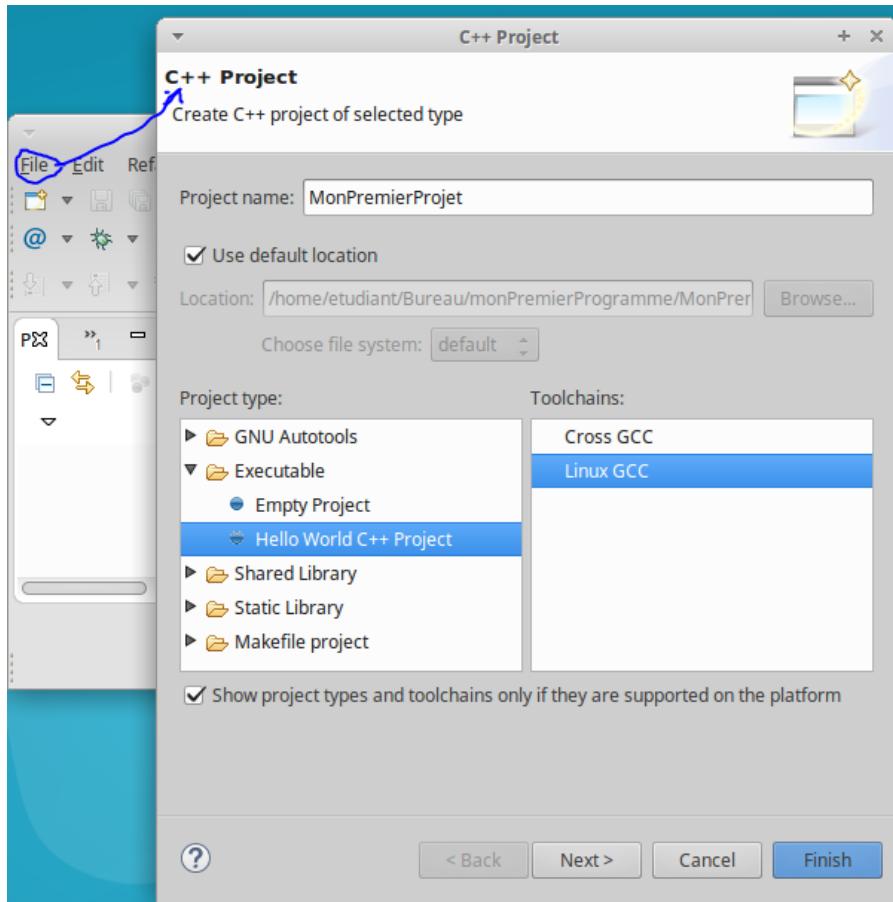


- Environnement configurable
- Répertoire «racine» de l'espace

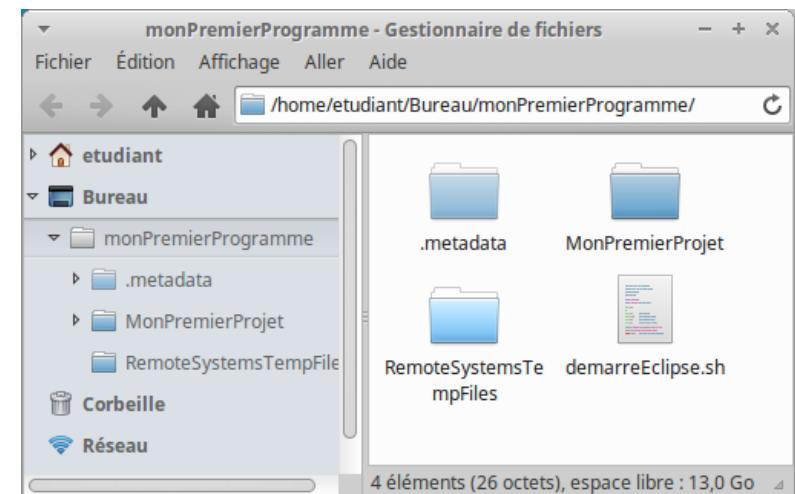




# Projet



- Une partie de l'espace de travail
- Option de compilation
- Réutilisation
- ...





# Programme exécutable

- La fonction main()

The screenshot shows the Eclipse C/C++ IDE interface. The title bar reads "C/C++ - MonPremierProjet/MonPremierProjet.cpp - Eclipse". The menu bar includes File, Edit, Refactor, Source, Navigate, Search, Project, Run, Window, and Help. The toolbar has various icons for file operations like Open, Save, and Build. The left sidebar shows the "Project" view with "MonPremierProjet" selected, displaying its structure with "Includes" and "MonPremierProjet.cpp" files. The central editor window displays the code for "MonPremierProjet.cpp":

```
1 //=====
2 // Name      : MonPremierProjet.cpp
3 // Author    : moi
4 // Version   :
5 // Copyright : Your copyright notice
6 // Description : Hello World in C++, Ansi-style
7 //=====
8
9 #include <iostream>
10 using namespace std;
11
12 int main()
13 {
14     cout << "!!!mon premier programme!!!" << endl; // print
15     return 0;
16 }
17
```

The right margin shows code completion suggestions for "iostream", "std", and "main() : int". Below the editor, the "Console" tab is active with the message "No consoles to display at this time." The status bar at the bottom shows "MonPremierProjet".



# Compilation et édition de liens

C/C++ - MonPremierProjet/MonPremierProjet.cpp - Eclipse

File Edit Refactor Source Navigate Search Project Run Window Help

Project MonPremierProjet

MonPremierProjet.cpp

```
1 //=====
2 // Name      : MonPremierProjet.cpp
3 // Author    : moi
4 // Version   :
5 // Copyright : Your copyright notice
6 // Description: Hello World in C++, Ansi-style
7 //=====

8
9 #include <iostream>
10 using namespace std;
11
12 int main()
13 {
14     cout << "!!!mon premier programme!!!" << endl; // print
15     return 0;
16 }
```

Includes

Debug

CDT Build Console [MonPremierProjet]

```
17:28:02 **** Build of configuration Debug for project MonPremierProjet ****
make all
Building file: ../MonPremierProjet.cpp
Invoking: GCC C++ Compiler
g++ -std=c++1y -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"MonPremierProjet.d" -MT"MonPremierProjet.o" -o "MonPremierProjet.o" "../MonPremierProjet.cpp"
Finished building: ../MonPremierProjet.cpp

Building target: MonPremierProjet
Invoking: GCC C++ Linker
g++ -o "MonPremierProjet" ./MonPremierProjet.o
Finished building target: MonPremierProjet
```

- Compilateur : g++
- Éditeur de lien : g++
- « sur fichier »

CDT Build Console [MonPremierProjet]

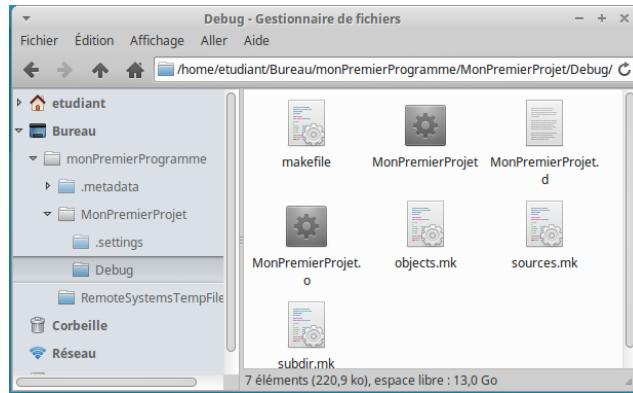
```
17:28:02 **** Build of configuration Debug for project MonPremierProjet ****
make all
Building file: ../MonPremierProjet.cpp
Invoking: GCC C++ Compiler
g++ -std=c++1y -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"MonPremierProjet.d" -MT"MonPremierProjet.o" -o "MonPremierProjet.o" "../MonPremierProjet.cpp"
Finished building: ../MonPremierProjet.cpp

Building target: MonPremierProjet
Invoking: GCC C++ Linker
g++ -o "MonPremierProjet" ./MonPremierProjet.o
Finished building target: MonPremierProjet
```

17:28:03 Build Finished (took 326ms)

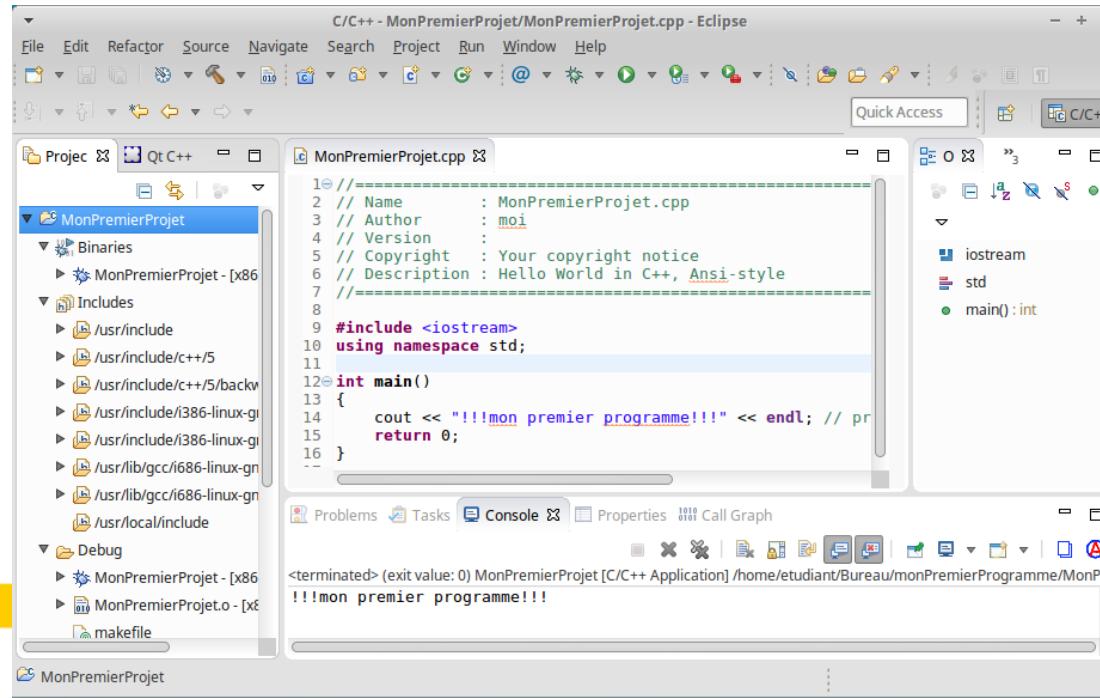


# Exécution



- En mode commande

- Directement à partir de l'environnement de développement



# Synthèse

---

- Programme
- Langages de programmation
- Compilateur
- Environnement de développement
- Mon premier programme

# **Programmation avancée en C++**

## **GIF-1003**

### **Module 1 : Codage des algorithmes**

1.3 Normes de programmation

T. Eude



UNIVERSITÉ  
**LAVAL**

Département d'informatique  
et de génie logiciel

## A lire

---

- Document sur les normes de programmation à appliquer dans le cadre du cours
- (voir site Web du cours, dans notes de cours)