

## Programmation orientée objet : Le langage C++

### OBJECTIFS

À l'issue de cette séquence, vous devrez être capable de :

- De créer une classe à partir d'un diagramme UML.
- De réaliser des entrées / sorties en mode console en utilisant les flux.
- D'interagir avec un fichier en utilisant un flux.
- D'instancier un objet, de manière automatique, dynamique et statique.
- D'appeler ses méthodes et de leur passer des paramètres en entrée et/ou en sortie
- De mettre en œuvre un passage de paramètres par valeur, par adresse, par référence.
- De différencier les différentes relations dans un diagramme de classes et de savoir les implémenter en C++.
- De définir des méthodes virtuelles et des classes abstraites.
- De comprendre le concept de template
- D'utiliser la librairie STL en particulier ses conteneurs et ses itérateurs associés.

Niveau de maîtrise attendu pour le BTS Systèmes Numérique option Informatique et Réseaux :

S4. Développement logiciel		IR
S4.6. Programmation orientée objet (Support : C++)	Du C au C++ : références, entrées/sorties (iostream, fstream), polymorphisme, etc.	3
	Définition de classes (encapsulation) et modèle canonique (dit de Coplien)	3
	Instanciation d'objets (new, delete, etc.)	4
	Surcharges d'opérateurs (injection, etc.)	2
	Mécanisme d'héritage	4
	Mécanismes d'agrégation et de composition	4
	Classes abstraites, virtualité	3
	Programmation générique : structure de la STL, conteneurs et itérateurs	2
	Programmation générique : classes paramétrées (template)	2
	Programmation générique : patrons de développement (design patterns)	1

# Sommaire

<b>Développement logiciel.....</b>	<b>3</b>
Programmation orientée objet : Le langage C++.....	3
<b>1. Du C au C++.....</b>	<b>3</b>
1.1. Objets et classes.....	3
1.2. Classification des objets.....	3
1.3. Espace de nommage.....	3
1.4. Type booléen.....	4
1.5. Surcharge ou polymorphisme de traitement.....	4
1.6. Entrées / Sorties : les flux.....	5
1.6.1. Généralités.....	5
1.6.2. Flux standards.....	5
1.6.3. Les manipulateurs de flux d'entrée-sortie.....	7
1.6.4. Les flux basés sur des fichiers.....	9
1.7. Références.....	13
1.8. Constantes en C++.....	14
1.9. Allocation et restitution de la mémoire.....	15
1.10. Traitement des erreurs : Les exceptions.....	15
1.10.1. Introduction.....	15
1.10.2. Traitement des erreurs.....	16
1.11. Conversion de types.....	17
<b>2. Définition de classes.....</b>	<b>18</b>
2.1. Introduction.....	18
2.2. Notion d'encapsulation.....	18
2.3. Modèle canonique dit Coplien.....	22
2.4. Application.....	25
2.5. Traitement des erreurs avec une Classe d'exception.....	27
<b>3. Relations entre objets.....</b>	<b>31</b>
3.1. Mécanisme d'héritage.....	31
3.1.1. Introduction.....	31
3.1.2. Modalités d'accès aux membres de la classe de base.....	33
3.1.3. Dérivations publique, protégée et privée.....	34
3.2. Mécanisme de composition.....	36
3.2.1. Introduction.....	36
3.2.2. Implémentation en C++.....	36
3.3. Mécanisme d'agrégation.....	40
3.3.1. Introduction.....	40
3.3.2. Implémentation en C++.....	40
3.4. Mécanisme d'association.....	43
3.4.1. introduction.....	43
3.4.2. Implémentation en C++.....	43
3.5. Notion de dépendance.....	46
<b>4. Fonction virtuelle, polymorphisme, classe abstraite.....</b>	<b>47</b>
4.1. Introduction.....	47
4.2. Fonctions virtuelles.....	49
4.3. Avantage des fonctions virtuelles.....	50
4.4. Fonctions virtuelles pures et Classes abstraites.....	50

4.5. Application.....	51
<b>5. Programmation générique : les « templates ».....</b>	<b>52</b>
5.1. Introduction.....	52
5.2. Fonctions modèles ou « template ».....	52
5.3. Les classes templates.....	54
<b>6. Programmation générique : Utilisation de la STL.....</b>	<b>56</b>
6.1. Introduction.....	56
6.2. Les conteneurs.....	56
6.3. Les itérateurs.....	57
6.4. Les algorithmes de la librairie STL.....	59
<b>7. Programmation générique : patrons de développement.....</b>	<b>60</b>
<b>8. Conclusion.....</b>	<b>60</b>