

Introduction

***Les compétences
en NSI***

L'enseignement de spécialité de numérique et sciences informatiques en terminale voie générale a pour objectif de poser les fondements de l'informatique afin de se préparer à une poursuite d'études dans l'enseignement supérieur.

Cet enseignement permet de développer les compétences suivantes :

- **analyser** et **modéliser** un problème en termes de flux et de traitement d'informations ;
- **décomposer** un problème en sous-problèmes, reconnaître des situations déjà analysées et réutiliser des solutions ;
- **concevoir** des solutions algorithmiques ;
- **traduire** un algorithme dans un langage de programmation, en spécifier les interfaces et les interactions, comprendre et réutiliser des codes sources existants, développer des processus de mise au point et de validation de programmes ;
- **mobiliser** les concepts et les technologies utiles pour assurer les fonctions d'acquisition, de mémorisation, de traitement et de diffusion des informations ;
- **développer** des capacités d'abstraction et de généralisation.

1 Les compétences

A Analyser et modéliser un problème en termes de flux et de traitement d'informations

Il s'agit d'appréhender les informations utiles à la résolution d'un problème ainsi que leur origine et le moyen de les traiter. Puis de mettre en œuvre les moyens afin de produire une solution au problème. Au préalable, il faut formaliser le problème, c'est-à-dire à expliciter dans un format technique (algorithme, langage, schéma...). Selon les situations, le choix du format ou l'outil utilisé sera à la charge de l'élève.

Conseils du professeur

Il faut toujours, dans un premier temps, s'appropriier le contexte, lire la totalité du sujet, parcourir les annexes disponibles, pour avoir une vision globale de la problématique avant de débiter toute phase de formalisation. Ensuite, il ne faut pas hésiter à prendre un papier brouillon et d'y mettre quelques idées avant de résoudre un problème sur ordinateur. Ceci permet d'organiser vos idées et vous fera gagner du temps par la suite.

B Décomposer un problème en sous-problèmes, reconnaître des situations déjà analysées et réutiliser des solutions

Un problème à résoudre peut être assez vaste au premier abord et donc un peu déroutant. Une astuce consiste à décomposer le problème en plusieurs sous-problèmes que l'on peut résoudre plus facilement. On retombera souvent sur des problèmes déjà résolus par le passé ou tout au moins sur des situations de départ similaires à certaines vues par le passé.

Conseils du professeur

Avant de débiter une conception ou une réalisation, nous devons faire un état des lieux de l'existant : qu'avons-nous à notre disposition ? Quelles sont les technologies disponibles dans ce contexte ? Sur quel existant puis-je me baser ?

C Concevoir des solutions algorithmiques

Un algorithme est une méthode précise constituée d'une **suite d'instructions logiques (d'étapes)** et permettant de résoudre un problème. Il s'agit finalement d'une liste d'étapes à suivre pour parvenir à une solution. C'est un peu comme une recette de cuisine ou un protocole opératoire en laboratoire. La conception va permettre de produire une solution qui corresponde parfaitement à la problématique, de manière optimisée. Il est nécessaire de connaître des méthodes

Remarque

Il existe bien souvent plusieurs algorithmes qui mènent à la même solution.

Conseils du professeur

Il est nécessaire de bien connaître les méthodes disponibles dans chaque domaine : méthodes d'analyse pour le développement, normes pour les infrastructures, conventions, réglementation...

Il est impossible de se dispenser de cette phase de conception.

D Traduire un algorithme dans un langage de programmation, en spécifier les interfaces et les interactions, comprendre et réutiliser des codes sources existants, développer des processus de mise au point et de validation de programmes

Une fois l'algorithme écrit, il doit être **traduit en une suite d'instructions comprises par une machine**, (comme un ordinateur, ou une calculatrice). La machine sera alors à même d'effectuer les calculs nécessaires qui mènent à la solution du problème étudié. Cette étape de traduction s'intitule la **programmation de l'algorithme**.

Une fois le programme écrit, il est essentiel de vérifier sa validité, c'est-à-dire que le programme est correct et qu'il produit bien le résultat attendu.

Conseils du professeur

Nous avons dit qu'il existe bien souvent plusieurs algorithmes qui mènent à la même solution. Il en va de même pour leur programmation : il existe bien souvent plusieurs façons de programmer un même algorithme. Aussi, lorsque vous écrirez de programmes, il faudra bien garder en tête qu'il n'existe pas toujours de solution unique à un même problème.

E Mobiliser les concepts et les technologies utiles pour assurer les fonctions d'acquisition, de mémorisation, de traitement et de diffusion des informations

Dans le monde actuel, la récolte, le stockage et la diffusion de l'information est devenue une activité primordiale. Aussi, les entreprises utilisent elles-mêmes des **systèmes d'information** (appelés **SI**) qui sont justement des ensembles de ressources qui permet de collecter, stocker, traiter et distribuer de l'information, en général grâce à des ordinateurs.

Le programme de Numérique et Sciences Informatiques ne requiert pas la connaissance d'un système d'information en particulier mais plutôt la connaissance de quelques technologies permettant l'acquisition, la mémorisation, le traitement et la diffusion de données. Il faut savoir y faire appel à bon escient.

F Développer des capacités d'abstraction et de généralisation

L'écriture d'algorithmes et de programmes informatiques vous permettra de développer une forte capacité d'abstraction qui sera répercutée sur votre capacité à résoudre un problème aussi bien en mathématiques, qu'en physique, chimie, SVT etc. Vous devrez en effet bien souvent partir d'un problème très spécifique et développer une résolution plus abstraite en modélisant les variables par des lettres, les phénomènes par une formule mathématique.

Conseils du professeur

Cette démarche d'abstraction vous sera bien utile en cours de physique chimie où vous devrez bien souvent tâcher de parvenir à une expression littérale plutôt que d'effectuer votre raisonnement sur des valeurs numériques.

Compétences transversales

Au cours de ce livre, diverses activités sont proposées, celles-ci vous permettront de :

- Faire preuve d'autonomie, d'initiative et de créativité ;
- Présenter un problème ou sa solution, développer une argumentation dans le cadre d'un débat ;
- Coopérer au sein d'une équipe dans le cadre d'un projet ;
- Rechercher de l'information, partager des ressources ;
- Faire un usage responsable et critique de l'informatique.

2 Récapitulatif des exercices illustrant les compétences

Compétences	Exercices concernés
▶ Analyser et modéliser un problème en terme de flux et de traitement d'informations	<ul style="list-style-type: none"> ➤ Chap. 1 : 1.4, 1.5 ➤ Chap. 3 : 3.1, 3.2 ➤ Chap. 4 : 4.1 ➤ Chap. 5 : 5.1, 5.2, 5.3 ➤ Chap. 6 : 6.5, 6.6, 6.7 ➤ Chap. 7 : 7.1, 7.2, 7.3 ➤ Chap. 8 : 8.2, 8.4, 8.5, 8.6 ➤ Chap. 9 : 9.1, 9.2 ➤ Chap. 11 : 11.3, 11.4 ➤ Chap. 12 : 12.1, 12.2
▶ Décomposer un problème en sous-problèmes, reconnaître des situations déjà analysées et réutiliser des solutions.	<ul style="list-style-type: none"> ➤ Chap. 1 : 1.3, 1.5 ➤ Chap. 2 : 2.1, 2.3 ➤ Chap. 3 : 3.4, 3.6 ➤ Chap. 5 : 5.3, 5.4 ➤ Chap. 8 : 8.1, 8.4, 8.5 ➤ Chap. 9 : 9.3 ➤ Chap. 10 : 10.2 ➤ Chap. 12 : 12.2
▶ Concevoir des solutions algorithmiques	<ul style="list-style-type: none"> ➤ Chap. 4 : 4.1 ➤ Chap. 6 : 6.1, 6.2, 6.3, 6.4 ➤ Chap. 7 : 7.4 ➤ Chap. 9 : 9.3, 9.4 ➤ Chap. 12 : 12.1, 12.3, 12.4

Compétences	Exercices concernés
▶ Traduire un algorithme dans un langage de programmation, spécifier les interfaces et les interactions, développer des processus de mise au point et de validation de programmes	<ul style="list-style-type: none">▶ Chap. 1 : 1.6▶ Chap. 4 : 4.1, 4.2▶ Chap. 5 : 5.7▶ Chap. 6 : 6.1, 6.2, 6.3, 6.4▶ Chap. 10 : 10.1, 10.3, 10.4, 10.5▶ Chap. 12 : 12.5
▶ Mobiliser les concepts et les technologies utiles pour assurer les fonctions d'acquisition, de mémorisation, de traitement et de diffusion des informations	<ul style="list-style-type: none">▶ Chap. 2 : 2.2, 2.5▶ Chap. 3 : 3.3▶ Chap. 5 : 5.1, 5.2, 5.5▶ Chap. 7 : 7.1, 7.2, 7.3▶ Chap. 8 : 8.1, 8.2▶ Chap. 9 : 9.4▶ Chap. 10 : 10.5▶ Chap. 11 : 11.1, 11.2
▶ Développer des capacités d'abstraction et de généralisation	<ul style="list-style-type: none">▶ Chap. 1 : 1.1, 1.2▶ Chap. 2 : 2.4▶ Chap. 3 : 3.5▶ Chap. 4 : 4.2▶ Chap. 5 : 5.1, 5.5▶ Chap. 8 : 8.3, 8.7▶ Chap. 9 : 9.1, 9.2▶ Chap. 10 : 10.1, 10.3▶ Chap. 12 : 12.4

Partie 1

Structures de données

