

# MICRO-INFORMATIQUE

## DÉFINITION DU DOMAINE D'EXAMEN

PROGRAMMATION

INF-5063-2

# **MICRO-INFORMATIQUE**

## **DÉFINITION DU DOMAINE D'EXAMEN**

### **PROGRAMMATION**

**INF-5063-2**

© Gouvernement du Québec  
Ministère de l'Éducation, 1998 — 98-0285

ISBN 2-550-33224-5

Dépôt légal — Bibliothèque nationale du Québec, 1998

## **1 Présentation**

La présente définition du domaine d'examen a été rédigée à des fins d'évaluation sommative. Elle permet de décrire et d'organiser les éléments essentiels et représentatifs du programme d'études et, plus particulièrement, du cours *Programmation*. La définition est fondée sur le programme, mais elle ne peut, en aucun cas, le remplacer. Son rôle est d'assurer la correspondance entre le programme et les épreuves nécessaires à l'évaluation sommative.

Les sections de la présente définition du domaine d'examen sont semblables à celles des définitions du domaine d'examen des autres cours. Leur contenu, cependant, est particulier au présent cours.

Le but de la définition du domaine d'examen est de préparer des épreuves valides d'une version à une autre, d'une année à une autre, ou encore d'un organisme scolaire à un autre en tenant compte du partage des responsabilités entre le ministère de l'Éducation et les organismes scolaires.

## 2 Conséquences des orientations du programme en évaluation sommative

### **Orientations**

### **Conséquences**

Le cours *Programmation* a pour but de familiariser l'élève avec les concepts de base relatifs à la programmation.

On évaluera la capacité de l'élève à appliquer de façon autonome les éléments de base d'un langage de programmation.

Dans le programme, on privilégie une approche pragmatique.

L'évaluation devra permettre de mettre en évidence les habiletés pratiques de l'élève.

Les objectifs du cours se situent sur le plan de l'analyse, de l'organisation et de la production d'un programme informatique.

On évaluera le programme informatique de l'élève et sa capacité à analyser et à organiser sa production.

Dans le programme, on tient compte des langages de programmation les plus répandus.

L'évaluation sommative devra pouvoir s'adapter aux différents langages de programmation.

### 3 Contenu du cours aux fins de l'évaluation sommative

#### **Notions**

- Planification et analyse du programme :
  - Présentation et justification;
  - Validation des instructions;
  - Programme exécutable;
  - Améliorations ou modifications.
  
- Production du programme :
  - Éléments de programmation;
  - Algorithme;
  - Encodage.

#### **Habilités**

- **Analyser** : Rechercher les éléments d'une application, mettre en évidence les relations entre les éléments et l'usage fait de ceux-ci.
  
- **Produire** : Intégrer de façon pertinente, originale et organisée différentes habiletés et différents éléments dans la création d'un tout complexe.

**4 Tableau de pondération**

NOTIONS	PLANIFICATION ET ANALYSE DU PROGRAMME	PRODUCTION DU PROGRAMME
<b>HABILETÉS</b>	<b>35 %</b>	<b>65 %</b>
<b>ANALYSER</b>  35 %	<b>1</b> Présentation et justification  10 %	
	<b>5</b> Validation des instructions  5 %	
	<b>6</b> Programme exécutable  10 %	
	<b>7</b> Améliorations ou modifications  10 %	
<b>PRODUIRE</b>  65 %		<b>2</b> Éléments de programmation  30 %
		<b>3</b> Algorithme  10 %
		<b>4</b> Encodage  25 %

Note : Les dimensions sont numérotées dans l'ordre logique d'exécution des comportements observables.

## 5 Comportements observables

### Description générale

C'est à partir de la liste des comportements observables établie ci-dessous que doit être construit l'outil d'évaluation.

L'élève doit pouvoir :

1. Décrire les éléments, les étapes, les objectifs visés et les étapes de création d'un programme informatique.
2. Composer un programme comprenant **au moins**, les éléments de programmation suivants :
  - un positionnement;
  - une variable;
  - une procédure ou une fonction d'initialisation;
  - une boucle simple.
3. Rédiger l'algorithme du programme en respectant les règles ainsi que l'ordre d'exécution des procédures.
4. Effectuer l'encodage de l'algorithme du programme à l'aide d'un langage de programmation.
5. Valider les instructions du programme.
6. Démontrer que le programme est exécutable.
7. Proposer des améliorations ou des modifications au programme.



**6 Justification des choix**

La pondération des dimensions correspond à l'importance que nous leur accordons dans l'atteinte des objectifs du cours.

Les dimensions étant liées aux *Notions* et aux *Habilités*, la pondération de celles-ci découle de la pondération accordée aux dimensions.

**Répartition de la pondération**

- En fonction des notions :
  - Planification et analyse du programme 35 %
  - Production du programme 65 %
  
- En fonction des habiletés :
  - Habileté *Analyser* 35 %
  - Habileté *Produire* 65 %

Afin d'assurer le plus d'équité possible dans l'évaluation des dossiers, qui étant donné la nature même du cours seront très différents, nous avons établi une liste de critères devant servir à cette évaluation. De plus, nous avons déterminé les critères qui s'appliquent à chaque dimension et nous avons précisé la répartition de la pondération entre ces critères. On trouvera cette information à la section 7, *Spécification de l'épreuve*, du présent document.

## **7 Spécification de l'épreuve**

### **7.1 Type d'épreuve**

L'évaluation des compétences acquises par l'élève dans le cours *Programmation* se fait à partir d'un dossier préparé par l'élève et comportant les éléments suivants :

- La description des objectifs visés et des étapes de réalisation du programme informatique;
- L'algorithme du programme;
- Le programme lui-même;
- La description des améliorations ou des modifications suggérées.

Une grille d'évaluation servira à l'analyse des dossiers présentés.

### **7.2 Caractéristiques de l'épreuve**

- On doit s'assurer que l'élève connaît les conditions et les modalités de l'évaluation.
- Selon le langage de programmation utilisé, on déterminera l'ampleur du programme à produire.
- L'élève remet, à la fin du cours, le dossier décrit ci-dessus.
- On doit s'assurer que les différents éléments inclus dans le dossier ont été réalisés par l'élève et de façon autonome.
- L'usage d'un ordinateur est nécessaire pour la création du programme.
- La grille d'évaluation du dossier doit être construite à partir des comportements observables décrits pour chacune des dimensions.
- La grille d'évaluation du dossier doit respecter la pondération prévue dans le tableau de pondération.
- La grille d'évaluation du dossier doit respecter les critères d'évaluation énumérés à la section 7.3 ainsi que la répartition et la pondération présentées à la section 7.4.

### **7.3 Critères d'évaluation**

Les critères suivants seront utilisés pour évaluer les dossiers des élèves :

- **Pertinence** : Lien significatif entre les besoins déterminés et les résultats attendus.
- **Efficienc**e : Rapport entre l'utilité du résultat, les objectifs visés et les moyens mis en oeuvre pour obtenir le résultat.
- **Cohérence** : Logique des liens.
- **Précision** : Exactitude des actions.
- **Esthétique** : Qualité de la présentation, ergonomie.

### 7.4 Répartition de la pondération en fonction des critères d'évaluation

Le tableau suivant précise les critères qui doivent être utilisés pour mesurer chaque dimension.

DIMENSIONS	CRITÈRES	PERTINENCE	EFFICIENCE	COHÉRENCE	PRÉCISION	ESTHÉTIQUE
		32 %	35 %	20 %	8 %	5 %
1	Présentation et justification 10 %	√ 5 %	√ 5 %			
2	Éléments de programmation 30 %	√ 20 %	√ 10 %			
3	Algorithme 10 %		√ 5 %	√ 5 %		
4	Encodage 25 %		√ 10 %	√ 10 %	√ 5 %	
5	Validation des instructions 5 %		√ 2 %		√ 3 %	
6	Programme exécutable 10 %	√ 2 %	√ 3 %			√ 5 %
7	Améliorations ou modifications 10 %	√ 5 %		√ 5 %		

### 7.5 Exigence de réussite

- La note de passage de l'épreuve est fixée à 60 sur 100.

