

# RCP : Référentiel de compétences en programmation - vue formulaire

Voir la vue générale Voir la vue détaillée avec l'ensemble des tâches élémentaires Voir la documentation

Compétences	(E) Évaluer	(M) Modéliser	(A) Anticiper	(D) Décomposer	(G) Généraliser	(X) Abstraire
Tout sélectionner	Tout sélectionner	Tout sélectionner	Tout sélectionner	Tout sélectionner	Tout sélectionner	Tout sélectionner
<b>(01)</b> <b>Programmer avec des expressions</b> Tout sélectionner	(E01) <input type="checkbox"/> Évaluer la valeur et le type (entier, chaîne de caractères ou booléen) d'une expression comportant variables, constantes et opérateurs	(M01) <input type="checkbox"/> Modéliser les informations disponibles/à calculer par des variables de type élémentaire (entier, chaîne de caractères et booléen)	(A01) <input type="checkbox"/> Anticiper l'écriture d'une expression comportant variables, constantes et opérateurs	(D01) <input type="checkbox"/> Décomposer une expression complexe en identifiant une ou des variables intermédiaires	(G01) <input type="checkbox"/> Généraliser une expression en remplaçant une constante par une variable	
<b>(02)</b> <b>Programmer avec des affectations et des séquences</b> Tout sélectionner	(E02) <input type="checkbox"/> Évaluer les effets de l'exécution d'un programme comportant une séquence d'affectations	(M02) <input type="checkbox"/> Modéliser les informations à calculer par une ou des variables intermédiaires	(A02) <input type="checkbox"/> Anticiper l'écriture d'un traitement séquentiel pour obtenir un résultat spécifié, la méthode étant donnée	(D02) <input type="checkbox"/> Décomposer un traitement complexe en identifiant une ou des variables intermédiaires		
<b>(03)</b> <b>Programmer avec des boucles bornées</b> Tout sélectionner	(E03) <input type="checkbox"/> Évaluer les effets de l'exécution d'un programme comportant une boucle bornée	(M03) <input type="checkbox"/> Modéliser par une ou des variables les informations à modifier à chaque répétition	(A03) <input type="checkbox"/> Anticiper l'écriture d'une boucle bornée pour obtenir un résultat spécifié, la méthode étant donnée	(D03) <input type="checkbox"/> Décomposer un traitement complexe en identifiant une boucle bornée	(G03) <input type="checkbox"/> Généraliser un traitement séquentiel en le remplaçant par un traitement répétitif	
<b>(04)</b> <b>Programmer</b>	(E04) <input type="checkbox"/> Évaluer les effets de	(M04) <input type="checkbox"/> Modéliser par une ou des	(A04) <input type="checkbox"/> Anticiper l'écriture d'un	(D04) <input type="checkbox"/> Décomposer un		

## avec des boucles non bornées

Tout sélectionner

l'exécution d'un programme comportant une boucle non bornée

variables les informations à modifier à chaque répétition

traitement répétitif pour obtenir un résultat spécifié, la méthode étant donnée

traitement complexe en identifiant une boucle non bornée

## (05) Programmer avec des constructions alternatives ou conditionnelles

Tout sélectionner

(E05) ☐ Évaluer les effets de l'exécution d'un programme comportant une ou des constructions alternatives ou conditionnelles

(M05) ☐ Modéliser par une variable booléenne la condition d'une alternative ou d'une conditionnelle

(A05) ☐ Anticiper l'écriture d'un traitement alternatif ou conditionnel pour obtenir un résultat spécifié, la méthode étant donnée

(D05) ☐ Décomposer un traitement complexe en plusieurs cas en utilisant des constructions alternatives ou conditionnelles

## (06) Programmer avec des fonctions

Tout sélectionner

(E06) ☐ Évaluer le résultat de l'appel d'une fonction avec des paramètres donnés

(M06) ☐ Modéliser un traitement par une fonction en spécifiant ses paramètres

(A06) ☐ Anticiper l'écriture d'une fonction, les paramètres étant spécifiés

(D06) ☐ Décomposer un traitement complexe en utilisant la composition de fonctions

(G06) ☐ Généraliser un traitement, par la définition d'une fonction avec paramètres ou par l'ajout d'un paramètre à une fonction

(X06) ☐ Abstraire un traitement par une définition de fonction avec paramètres et commentaire

## (07) Programmer avec des fonctions récursives

Tout sélectionner

(E07) ☐ Évaluer le résultat de l'appel d'une fonction récursive

(M07) ☐ Modéliser un traitement par une fonction récursive en spécifiant ses paramètres

(A07) ☐ Anticiper l'écriture d'une fonction récursive, les paramètres étant spécifiés

(G07) ☐ Généraliser une fonction récursive avec une fonction en paramètre

## (08) Programmer avec des tuples

Tout sélectionner

(E08) ☐ Évaluer une expression permettant d'accéder aux informations d'un tuple

(M08) ☐ Modéliser une information complexe par un tuple d'informations élémentaires

(A08) ☐ Anticiper l'accès aux informations élémentaires d'un tuple

(X08) ☐ Abstraire des données complexes en utilisant des tuples

## (09) Programmer

(E09) ☐ Évaluer une expression permettant

(M09) ☐ Modéliser une information complexe par un

(A09) ☐ Anticiper l'accès aux informations

(X09) ☐ Abstraire des données complexes en

## avec des enregistrements à champs nommés

Tout sélectionner

d'accéder aux informations d'un enregistrement à champs nommés

enregistrement à champs nommés

élémentaires d'un enregistrement à champs nommés

utilisant des enregistrements à champs nommés

## (10) Programmer avec des tableaux - construction et accès

Tout sélectionner

(E10) ☐ Évaluer une expression ou les effets de l'exécution d'un programme utilisant un tableau

(M10) ☐ Modéliser une série d'informations par un tableau donné en extension

(A10) ☐ Anticiper les traitements à programmer pour accéder aux éléments d'un tableau par leurs indices et les modifier

(G10) ☐ Généraliser l'écriture d'un tableau en utilisant la compréhension

(X10) ☐ Abstraire les données complexes en utilisant des tableaux

## (11) Programmer des parcours de tableau

Tout sélectionner

(E11) ☐ Évaluer les effets de l'exécution d'un programme itérant sur un tableau

(M11) ☐ Modéliser un traitement de tableau en choisissant d'itérer sur les indices ou les éléments

(A11) ☐ Anticiper un parcours linéaire de tableau

(G11) ☐ Généraliser un traitement en l'appliquant à tous les éléments d'un tableau

## (12) Programmer avec des dictionnaires - construction et accès

Tout sélectionner

(E12) ☒ Évaluer une expression ou les effets de l'exécution d'un programme utilisant un dictionnaire

(M12) ☐ Modéliser l'association d'informations par un dictionnaire

(A12) ☐ Anticiper le traitement permettant d'ajouter ou modifier une association à un dictionnaire

(X12) ☐ Abstraire les données complexes en utilisant des dictionnaires

## (13) Programmer avec des dictionnaires - itérer

Tout sélectionner

(E13) ☐ Évaluer les effets de l'exécution d'un programme itérant sur un dictionnaire

(M13) ☐ Modéliser un traitement de dictionnaire en choisissant d'itérer sur les clés, les valeurs ou les paires

(A13) ☐ Anticiper un traitement nécessitant d'itérer sur un dictionnaire

## (14) Programmer avec des

(M14) ☐ Modéliser les données et les traitements en

(A14) ☐ Anticiper l'implémentation

(X14) ☒ Abstraire un traitement en utilisant une

interfaces et des implémentations

Tout sélectionner

(15) Programmer avec des objets et des classes

Tout sélectionner

(E15) ☐ Évaluer la création d'une instance, l'invocation de méthodes et l'accès aux attributs d'un objet

spécifiant une structure de données par son interface

(M15) ☐ Modéliser les données et les traitements en spécifiant une classe, des attributs et des méthodes

d'une structure de données

(A15) ☐ Anticiper des traitements en implémentant les méthodes d'une classe

(D15) ☐ Décomposer les données et les traitements en les regroupant dans des classes

(G15) ☐ Généraliser une classe par l'ajout d'un attribut ou d'une méthode

structure de données à spécifier.

(X15) ☐ Abstraire un ensemble de données et de traitements en les encapsulant dans une classe en tant qu'attributs et méthodes

Copier la liste des codes des capacités sélectionnées

Télécharger un fichier en markdown avec une tâche par capacité sélectionnée

Télécharger un fichier en markdown avec toutes les tâches pour chaque capacité sélectionnée

Sophie Chane-Lune, Sébastien Hoarau, Christophe Declercq, Laboratoire d'Informatique et de Mathématiques, Université de La Réunion, 2023-2025