

- Faculté des sciences économiques
- www.unine.ch/seco

Programmation (5MI1003)

Filières concernées	Nombre d'heures	Validation	Crédits ECTS
Bachelor en mathématiques	Cours: 2 ph TP: 2 ph	Voir ci-dessous	6
Bachelor en sciences économiques, orientation économie	Cours: 2 ph TP: 2 ph	Voir ci-dessous	6
Bachelor en sciences économiques, orientation économie politique	Cours: 2 ph TP: 2 ph	Voir ci-dessous	6
Bachelor en sciences économiques, orientation management	Cours: 2 ph TP: 2 ph	Voir ci-dessous	6

ph=période hebdomadaire, pg=période globale, j=jour, dj=demi-jour, h=heure, min=minute

Période d'enseignement:

- Semestre Automne

Equipe enseignante

Adrian Holzer
Professeur
Institut du management de l'information
A.L.Breguet 2,
CH 2000 Neuchâtel +41.32.718.13.32
adrian.holzer@unine.ch

Aditya Purohit
Assistant-doctorant
Institut du management de l'information
A.L.Breguet 2, CH-2000 NEUCHATEL
aditya.purohit@unine.ch

Kristoffer Bergram
Assistant-doctorant
Institut du management de l'information
A.L.Breguet 2, CH-2000 NEUCHATEL +41327181572
kristoffer.bergram@unine.ch

Contenu

Pensée computationnelle (modélisation, algorithmique)
Bases de programmation en langage Python (variables, structures de données, fonctions, classes, objets)

Forme de l'évaluation

Travaux pratiques individuels (30%)
Présentation du travail d'équipe (30%)
Rapport de travail d'équipe (40%)

Modalités de rattrapage: examen écrit en session d'examen (100%)

Les objets connectés ne sont pas admis dans les examens. En cas d'infraction aux règles, les étudiants sont considérés en situation de fraude et les éléments non admis seront retirés. En outre, l'examen pourra être réputé échoué.

Documentation

N. Bourgeois (2017), Python du grand débutant à la programmation objet, Ellipses.
A. Sweigart (2017), Invent Your Own Computer Games with Python, No starch press

- Faculté des sciences économiques
- www.unine.ch/seco

Programmation (5MI1003)

Pré-requis

Aucun

Forme de l'enseignement

2h cours + 2h TP

Objectifs d'apprentissage

Au terme de la formation l'étudiant-e doit être capable de :

- Produire une solution sous forme algorithmique
- Formuler un problème de manière computationnelle

Compétences transférables

- Stimuler sa créativité
- Gérer des projets
- Stimuler sa créativité
- Appliquer ses connaissances à des situations concrètes