

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

Software Design (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 durabilité	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 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

Dr Vladimir Macko
Lecturer
Institut du management de l'information
A.L. Breguet 2
CH-2000 Neuchâtel
vladimir.macko@unine.ch

Natalia Bartlomiejczyk
Assistante-doctorant
Institut du management de l'information
A.L.Breguet 2
CH-2000 Neuchâtel
Natalia.Bartlomiejczyk@unine.ch

Contenu

Computational thinking (decomposition, patterns, algorithms)
Programming with Python (instructions, conditions, loops, dictionaries, classes, inheritance)

Forme de l'évaluation

Individual Laboratory exercises during semester (2%)
Project code (10%)
Project presentation (18%)
Individual Two-hour written exam during the session without documentation (70%)

(Students are required to adhere to the AI policy specified for this class)

Modalités de rattrapage

Two-hour written individual exam during the exam session without documentation (100%)

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

K. Beecher (2019), Computational Thinking, a beginner's guide to problem solving and programming

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

Software Design (5MI1003)

Pré-requis

Technologies de l'information

Forme de l'enseignement

2h cours + 2h TP

Langue d'enseignement: anglais

Objectifs d'apprentissage

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

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

Compétences transférables

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