

CPNITS [p] – Creative Python Programming for Non-IT Students – 15 ECs

Module book

2022-2023

Version 1.6 / 12-01-2023



hogeschool

Creative Python Programming for Non-IT Students – 15 ECs

Version 1.6
12-01-2023

Study year 2022-2023

Module coordinator	Lecturers
Niels Doorn niels.doorn@nhlstenden.com	Victor G.B. Peters victor.peters@nhlstenden.com Jaqueline Berghout jaqueline.berghout@nhlstenden.com

Preface

In 2022, programming is no longer something you can let someone else do. The programmer's profession has caught up with its own obscurity. We now live in a time when everything revolves around IT, automated processes and program code. More and more applications offer possibilities to automate actions and these possibilities are gradually becoming unavoidable for the end user.

In short: everyone who has graduated from university in recent decades will sooner or later end up back in school to acquire one unavoidable skill: programming.

Before you get all sorts of terrifying images of sweat-smelling, Red Bull-drinking hoodies behind intricate pieces of code in creepy black screens; programming is a creative profession to begin with. Programming is puzzling with starting points and desired results. The scary code is just a trick that comes after that. Moreover, you will soon discover that that scary code is just English, which can be read and written like a story.

This brings us to the core of this minor: Every university student is welcome and can successfully complete this minor. Regardless of previous education or chosen study, every profile, every student, with the exception of Computer Science and ICT students.

The study load for this module is 420 hours (15 EC).

The various methods applied at NHL Stenden University of Applied Sciences have shown in the past that the method of the module "Creative Programming for Non-IT Students" is the most suitable method for teaching programming skills. Through student evaluations, the program has been adapted in such a way that more time is available for processing the acquired cognitive knowledge.

Assessment is based on a portfolio of assignments that is built up during this minor.

Niels Doorn & Victor Peters

Emmen, 01-09-2022

Contents

Preface	3
1 Introduction	6
1.1 Competences	7
1.2 Module theme.....	8
1.3 Module goals.....	8
1.4 Foreknowledge.....	8
1.5 Conventions.....	8
1.6 Version control	9
1.7 Attendance guest lecture, excursion or workshop.....	9
2 Reviewing	10
2.1 Assessment of assignments, portfolio	11
2.2 Module resit	11
3 Program	12
3.1 Lecture formats	12
3.1.1 Atelier	12
3.1.2 Fully online.....	12
3.2 Program overview	13
3.3 Weekly schedule	14
4 Structure & Organisation	23
5 Literature / software	24
5.1 Compulsory course material.....	24
5.2 Advised professional literature.....	24
5.3 Reference books.....	24
5.4 Software	24

6	Module evaluation	25
7	Appendices.....	26
7.1	Appendix 1: Scoring Rubrics	26

1 Introduction

There is no field in the world imaginable without an IT component. IT applications are used at every level in business, education or government, often consciously, often unconsciously. The modern professional must not only be able to understand how such an application 'thinks', but often must also be able to provide applications with additional and specific commands. General programming skills are indispensable for this.

Typical professional situation

Gijs is a safety manager at the Municipality of Amsterdam. During major events, pedestrian and traffic flows must be monitored but can also be influenced on an ad hoc basis. For example, if a calamity occurs.

A specific application Geographic Information System (GIS) is used to map all this movement data. Based on measurements and models, this software can make suggestions about possible closures and diversions during an event.

Those suggestions are of course different for every area, for every event and even for every weather type. These specific properties and the specific wishes of the security services on the street can be defined in a Python programming environment.

In order to be able to deliver this type of customization permanently, Gijs, as a safety expert, must be able to provide the application with extra intelligence at any time using the Python and Java programming languages. A situation where one is dependent on an external programmer is unworkable and undesirable.

During his studies, Gijs learned to program Integral Security, among others with the programming languages C and Python. Thanks to this skill, he can immediately start working with the GIS software for his work. Learning the Java programming language also took little effort thanks to the acquired skills. An external programmer was therefore not necessary.

1.1 Competences

Within this module, as a starting professional, you will work on a number of competencies related to analysis, design and realization.

	Manage	Analyse	Advise	Design	Realize
User Interaction				Translating advice into a simple user interaction with a standard prototyping technique.	Realization and qualitative testing of simple interactive products or services based on an interaction design using common tools, design guidelines and/or corporate identity.
Processes					
Infrastructure					
Software		Collect and validate functional requirements for a software system with one stakeholder according to a standard method.		Making a design for a software system, including database, with modelling techniques according to a standard method.	Building, testing and making available a simple software system. Setting up, filling and querying a database is part of the software system.
Hardware interfacing					

1.2 Module theme

Within this module, the student learns various theoretical and practical knowledge in the field of programming and databases. This is done with video lessons, personal guidance and practical assignments. Learning takes place as much as possible on the basis of practical assignments.

1.3 Module goals

After successfully completing this module, the student will be able to:

- > Read, understand and create applications in Python programming language;
- > Working with programming structures such as: repetitions, conditional command blocks, functions, multiple data structures;
- > Insight into personal development as a programmer within one's own field;
- > Insight into the way in which the student can master a new programming language on the basis of the knowledge and skills acquired in this minor.

1.4 Foreknowledge

No specific prior knowledge is required to follow this module.

1.5 Conventions

The conventions to be used with regard to naming and classification of the program code are part of the course material.

1.6 Version control

Version	Date	Auteur	Description
1.0	28-05-2020	V. Peters	Initiële versie
1.1	11-11-2020	V. Peters	Volledig online aangepast
1.2	03-02-2020	V. Peters	Aanpassing aan CS50 2021
1.3	25-08-2021	V. Peters	Aanpassing Assignments en afronding
1.4	04-11-2021	V.G.B. Peters	Correcties en aanpassing nieuwe CS50
1.5	23-08-2022	V.G.B. Peters	Aanpassing volgorde van Assignments
1.6	12-01-2023	J.S. Berghout	Aanpassingen Assignments

1.7 Attendance guest lecture, excursion or workshop

This module is offered by the student in the form of home study. During two days a week, the student is expected in the online studio at certain times. This is mandatory. During the home study days, the student must be available online for teachers and for group assignments with other students.

Active presence at the workshops is mandatory. In the event of insufficient (less than 80%) attendance, the student is regarded as not assessable. A resit is therefore not possible.

2 Reviewing

In order to assess whether you have achieved the objectives of this module, all assignments to be submitted are placed in a portfolio, with a number of points of attention for accountability. At the end of the module, the final result of this minor is determined on the basis of the portfolio containing the assignments.

The reviewing will take place by means of an assessment form (Scoring Rubrics), which can be found in Appendix 1. This module is assessed with a pass if the standard has been met. The standard is determined using the scoring rubrics from Appendix 1.

2.1 Assessment of assignments, portfolio

During this module, the student works on a portfolio that can be viewed online by the teacher throughout the entire period. The completed assignments and their feedback from CS50 Submit are also placed in this portfolio.

At the end of the module, the portfolio is handed in digitally, containing all completed assignments, accompanied by comments and considerations as stated in the Weekly Programmes.

This portfolio shows sufficient comments and conclusions to determine to what extent the student has fully carried out and understood the assignments, and how the student makes the relationship with his own field.

So, in this portfolio you can find:

- For every week
 - Completed CS50 assignments
 - Programming code
 - Screenshot of the Submit results
 - URL to the Submit results
 - Notes in accordance with the weekly assignments as elaborated later in this module book
 - Results and evaluations of lecture activities
- The end project and your evaluation on the end project

2.2 Module resit

Students who do not pass can do one resit.

3 Program

This module includes several weekly labs during atelier time. During these lectures, the student can discuss problems with the assignments with the teachers and fellow students. In addition to the studios, plenty of home study will have to be done. It is expected that 3/5 of the total study time consists of home study, with teachers available for questions and assistance.

The workshops are prepared by studying the specified subject matter prior to the workshop. Assignments can be made in your own time and in the studio. Some assignments are individual, others can be done in a team.

The elaborated commands should be created and tested in the CS50 IDE. The code is submitted via CS50 Submit, which automatically returns feedback and approval. The code and feedback from CS50 will be added to the portfolio. The way of handing in will be discussed during the first workshop.

3.1 Lecture formats

The different lecture formats are described below.

3.1.1 Atelier

The student prepares the workshops by means of self-study, so that optimal use can be made during these workshops of the presence of fellow students and teachers. A specific problem or aspect will be discussed regularly in class, for which the student has been able to make preparations.

3.1.2 Fully online

This module can be followed completely online.

3.2 Program overview

In the overview below you can find an overview of the weekly activities.

Week	Task. No.	Study activity
1.1	3.3.0	Functions, Variables Conditionals
1.2	3.3.1	Loops Exceptions
1.3	3.3.2	Libraries
1.4	3.3.3	Unit Tests
1.5	3.3.4	File I/O
1.6	3.3.5	Regular Expressions
1.7	3.3.6	Object-Oriented Programming
1.8	3.3.7	Final Project
1.8 + 1.9		Reviewing and closure

3.3 Weekly schedule

3.3.0 Introduction Computer technology + Functions, Variables & Conditionals

Week	1.1 CS50 0. Functions, Variables CS50 1. Conditionals
Work form	Atelier and self-study
Duration	2 x 8 hours Atelier 3 x 8 hours at home/online work and/or teamwork
Lesson Objectives	<ul style="list-style-type: none">• Kick-off of this module• Understanding how a computer works• Understanding how functions, variables and conditionals work• Simple programming with pseudocode• Data types
Contents	See title
Preparation	Study: https://cs50.harvard.edu/python/2022/weeks/0/ https://cs50.harvard.edu/python/2022/weeks/1/ Make sure you can work with your laptop in the CS50 IDE and that you can submit assignments with CS50 Submit.
Assignments	<ol style="list-style-type: none">1. Assignments:<ol style="list-style-type: none">a. Set up your work environment as describedb. Make the assignments in https://cs50.harvard.edu/python/2022/weeks/0/ https://cs50.harvard.edu/python/2022/weeks/1/2. Portfolio:<ol style="list-style-type: none">a. Create a portfolio document in Google Drive -> Google Document (Google's Word variant). Install the add-on Code Blocks. This allows you to paste code into a Doc and format it as code;b. Share this URL via a Discord Direct Message with your teacher.3. Describe in your portfolio:<ol style="list-style-type: none">a. Include the code from your CS50 Projects, along with screenshots of CS50 Submit feedback and the URL to this submit.

3.3.1 Loops, Exceptions

Week	1.2 CS50 2. Loops CS50 3. Exceptions
Work form	Atelier and self-study
Duration	2 x 8 hours Atelier 3 x 8 hours at home/online work and/or teamwork
Lesson Objectives	<ul style="list-style-type: none">• Understanding how loops work, dictionary• Learning to catch exceptions
Contents	See title
Preparation	Study: https://cs50.harvard.edu/python/2022/weeks/2/ https://cs50.harvard.edu/python/2022/weeks/3/ Study both the Lecture video and the Notes.
Assignments	<ol style="list-style-type: none">1. Assignments:<ol style="list-style-type: none">a. Make the Problem Sets at this week.b. Explain in comments what the different pieces of code mean and do.2. Portfolio:<ol style="list-style-type: none">a. Describe in your portfolio how you might find or apply what you have learned so far within your own field;b. Describe in your portfolio what you discovered and conclude about your way of learning to program;c. Include the code from your CS50 Projects, along with screenshots of CS50 Submit feedback and the URL to this submit.

3.3.2 Libraries

Week	1.3 CS50 4. Libraries
Work form	Atelier and self-study
Duration	2 x 8 hours Atelier 3 x 8 hours at home/online work and/or teamwork
Lesson Objectives	<ul style="list-style-type: none">• Working with APIs and JSON• Working with command-line arguments• Creating your own library• Understanding how to use the many pre-written python libraries
Contents	See title
Preparation	Study: https://cs50.harvard.edu/python/2022/weeks/4/ Study both the Lecture video and the Notes.
Assignments	<ol style="list-style-type: none">1. Assignments:<ol style="list-style-type: none">a. Make the Problem Sets at this week'sb. Explain in comments what the different pieces of code mean and do.2. Portfolio:<ol style="list-style-type: none">a. Describe in your portfolio how you might find or apply what you have learned so far within your own field;b. Describe in your portfolio what you discovered and conclude about your way of learning to program;c. Include the code from your CS50 Projects, along with screenshots of CS50 Submit feedback and the URL to this submit.

3.3.3 Unit Tests

Week	1.4 CS50 5. Unit Tests
Work form	Atelier and self-study
Duration	2 x 8 hours Atelier 3 x 8 hours at home/online work and/or teamwork
Lesson Objectives	<ul style="list-style-type: none">• Learning various ways of testing• Working with PyTest• Understanding what, how and why of testing• Building your own test environment
Contents	See title
Preparation	Study: https://cs50.harvard.edu/python/2022/weeks/5/ Study both the Lecture video and the Notes.
Assignments	<ol style="list-style-type: none">1. Assignments:<ol style="list-style-type: none">a. Make the Problem Sets at this week'sb. Explain in comments what the different pieces of code mean and do.2. Portfolio:<ol style="list-style-type: none">a. Describe in your portfolio how you might find or apply what you have learned so far within your own field;b. Describe in your portfolio what you discovered and conclude about your way of learning to program;c. Include the code from your CS50 Projects, along with screenshots of CS50 Submit feedback and the URL to this submit.

3.3.4 File I/O

Week	1.5 CS50 6. File I/O
Work form	Atelier and self-study
Duration	2 x 8 hours Atelier 3 x 8 hours at home/online work and/or teamwork
Lesson Objectives	<ul style="list-style-type: none">• Working with Pillow• Reading and writing text files• Exchanging data with CSV• The benefits of binary files
Contents	See title
Preparation	Study: https://cs50.harvard.edu/python/2022/weeks/6/ Study both the Lecture video and the Notes.
Assignments	<ol style="list-style-type: none">1. Assignments:<ol style="list-style-type: none">a. Make the Problem Sets at this week'sb. Explain in comments what the different pieces of code mean and do.2. Portfolio:<ol style="list-style-type: none">a. Describe in your portfolio how you might find or apply what you have learned so far within your own field;b. Describe in your portfolio what you discovered and conclude about your way of learning to program;c. Include the code from your CS50 Projects, along with screenshots of CS50 Submit feedback and the URL to this submit.

3.3.5 Regular Expressions

Week	1.6 CS50 7. Regular Expressions
Work form	Atelier and self-study
Duration	2 x 8 hours Atelier 3 x 8 hours at home/online work and/or teamwork
Lesson Objectives	<ul style="list-style-type: none">• Programming with regular expressions
Contents	See title
Preparation	Study: https://cs50.harvard.edu/python/2022/weeks/7/ Study both the Lecture video and the Notes.
Assignments	<ol style="list-style-type: none">1. Assignments:<ol style="list-style-type: none">a. Make the Problem Sets at this week'sb. Explain in comments what the different pieces of code mean and do.2. Portfolio:<ol style="list-style-type: none">a. Describe in your portfolio how you might find or apply what you have learned so far within your own field;b. Describe in your portfolio what you discovered and conclude about your way of learning to program;<ol style="list-style-type: none">a. Include the code from your CS50 Projects, along with screenshots of CS50 Submit feedback and the URL to this submit.

3.3.6 Object-Oriented Programming

Week	1.7 CS50 8. Object-Oriented Programming
Work form	Atelier and self-study
Duration	2 x 8 hours Atelier 3 x 8 hours at home/online work and/or teamwork
Lesson Objectives	<ul style="list-style-type: none">• Working with tuple and classes, objects and attributes• Various new data structures• Understanding inheritance• Object oriented programming• Working with special method names
Contents	See title
Preparation	Study: https://cs50.harvard.edu/python/2022/weeks/8/ Study both the Lecture video and the Notes.
Assignments	<ol style="list-style-type: none">1. Assignments:<ol style="list-style-type: none">a. Make the Problem Sets at this week'sb. Explain in comments what the different pieces of code mean and do.2. Portfolio:<ol style="list-style-type: none">a. Describe in your portfolio how you might find or apply what you have learned so far within your own field;b. Describe in your portfolio what you discovered and conclude about your way of learning to program;c. Include the code from your CS50 Projects, along with screenshots of CS50 Submit feedback and the URL to this submit.

3.3.7 Final Project

Week	1.8
Work form	Atelier and self-study
Duration	2 x 8 hours Atelier 3 x 8 hours at home/online work and/or teamwork
Lesson Objectives	<ul style="list-style-type: none">• Learn more about the world of Python Programming• Designing and building your own Python project
Contents	See title
Preparation	Study: https://cs50.harvard.edu/python/2022/weeks/9/ Study both the Lecture video and the Notes.
Assignments	<ol style="list-style-type: none">1. Assignments:<ol style="list-style-type: none">a. Create your own project as discussed during the workshops. Consult the outline of your Final Project with the instructor before building it. This project should reflect what you have learned and done over the past period and show a clear relationship to your own field of study. Describe what your project will do: that is, what the input will be, how it will be processed, and what the output will be. And describe why this project is relevant, what prompted you to create it.b. Also complete the assignment as described in CS50, so that this Final Project is also submitted there.2. Portfolio:<ol style="list-style-type: none">a. Describe in your portfolio how you might find or apply what you have learned so far within your own field;b. Describe in your portfolio what you discovered and conclude about your way of learning to program;c. Include the code from your CS50 Projects, along with screenshots of CS50 Submit feedback and the URL to this submit.

3.3.8 Assessment and closure

Week	1.8 + 1.9
Work form	Handing in products
Assignments	<ol style="list-style-type: none">1. Make sure your entire portfolio is in order at 11:59 PM on Wednesday of week 1.9.2. All assignments must be completed:<ol style="list-style-type: none">a. The programming code has commentsb. All code is pasted in the portfolioc. The feedback from CS50 Submit has been added as a screenshotd. The URL to the Check50 feedback has been added3. Each week contains a number with attention to the following questions:<ol style="list-style-type: none">a. How can you relate what you have learned to your own field?b. How did you learn business? What have you discovered about your way of learning?4. Your Final project is in the portfolio, as well as a working URL to the walk-through video about the final project.5. Final conclusion about IT in relation to your field and about the relationship you have built up with programming during this period.

4 Structure & Organisation

The schedule below provides an overview of all contact hours in this module.

In addition, students are expected to plan their own meetings where they can work on the assignments. This also applies to the time the student needs to prepare and complete (individual) assignments. This schedule also provides a good overview of the expected study load per student.

Table 4.1 Student contact hours (SCU) and Study load hours (SBU) per week:

Period 1	Week 1.1		Week 1.2		Week 1.3		Week 1.4		Week 1.5		Week 1.6		Week 1.7		Week 1.8		Week 1.9		Total	
Activity	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU
Atelier	16		16		16		16		16		16		16		16				128	0
Lecture																			0	0
Assessment																			0	0
Self-study		26		26		26		26		26		26		26		26		42	0	250
Totals	16	26	16	26	16	26	16	26	16	26	16	26	16	26	16	26	0	42	128	250
Extracurricular activities																			0	42
End total																				420

SCU = Student contact hours (60 minutes)

SBU = Study load hours (60 minutes)

5 Literature / software

Below is a list of the literature and software required to successfully complete and complete the course.

5.1 Compulsory course material

Online content <https://cs50.harvard.edu/python/2022/>

5.2 Advised professional literature

Book: [This is IT!](#), Van Duuren Media.

5.3 Reference books

- <https://www.python.org/doc>
- <https://www.w3schools.com/python>

5.4 Software

- Google Chrome-browser: <https://www.google.com/chrome>
- Google Documents: <https://docs.google.com/document/u/0>

6 Module evaluation

The module will be evaluated by means of a questionnaire at the end of the module. This questionnaire contains all parts of the module including organizational aspects, content, quality of teaching staff, etc.

We kindly ask you to participate in this evaluation. The results of this evaluation will be used to improve the next version of this module.

7 Appendices

7.1 Appendix 1: Scoring Rubrics

	Good: 9	Sufficient: 7	Not sufficient: 4	Weighing
Analysing the assignments	All assignments are well understood, and the execution of all assignments is in accordance with the assignment. Where relevant, the <i>more comfortable</i> assignments have always been carried out. The material offered can be found in the elaborations of the assignments.	Most assignments are well understood, and the execution of most assignments is in accordance with the assignment. The offered substance can be found in most of the elaborations	Several assignments were misunderstood, resulting in different elaborations than intended. In several assignments, the material offered has not been or has not been correctly incorporated in the elaborations of the assignments.	Should be at least <i>sufficient</i> . Must be <i>good</i> for an end score <i>good</i> .
Structural design	The available methods have clearly been used to structure the elaborations. The code is well documented.	Some available methods have been used to structure the elaborations. The code is sufficiently commented.	The majority of the assignments are not sufficiently worked out according to a standard method. The code is not or insufficiently provided with comments.	Should be at least <i>sufficient</i> . Must be <i>good</i> for an end score <i>good</i> .
End product and accountability	The products and the Final project work properly. Comments between the code explain how the delivered code functions.	Most products and the Final project work properly. Comments show understanding of the code created.	Many products or the Final project do not work or do not work properly. There are not enough comments to explain exactly how the delivered code works.	Should be at least <i>sufficient</i> . Must be <i>good</i> for an end score <i>good</i> .
Portfolio and accountability	The student gives a clear picture of the way in which the skills have been acquired and demonstrates that such skills can also be acquired and expanded through self-study outside of education.	The student gives a reasonable idea of how the skills have been acquired during this program and how they can be extended outside of education.	The student does not provide a clear picture of the way in which the skills have been acquired, or of a way in which skills can be acquired through self-study outside of education.	Should be at least <i>sufficient</i> . Must be <i>good</i> for an end score <i>good</i> .

