Portfolio Guidelines

Course: Object Oriented and Functional Programming with Python, DLBDSOOFPP01

Table of contents

1 Definition and objective of a portfolio

The academic assessment "Portfolio" is mainly a practical exercise that is designed and carried out individually. Nonetheless, this process is supervised by the responsible tutor.

In contrast to a written project report, this examination combines the practical implementation and development of a product with digital documentations as part of a portfolio.

The assessment corresponds to a reflection portfolio. The individual learning and examination process is recorded and accompanied in a reflective manner. Finally, there is the opportunity to use the portfolio within the framework of a presentation or career portfolio for application and "self-marketing".

The result of a portfolio always consists of a **"product"** in the actual or figurative sense. Depending on the module/course's specifications, this can consist of a software solution, a design, an installation, a process, a business plan or similar project.

The portfolio consists of three phases, which are intended to illustrate the individual work or development steps and the adopted approach. The learning progress or competence increase achieved in this multi-phase process is documented using three portfolio phases within a portfolio software called “PebblePad”. The three phases mentioned are the “conception phase”, the “development/reflection phase” and the “finalization phase”. These are explained in more detail in Chapter 2. The course-specific requirements of the respective phases and portfolio pages form parts of the final portfolio. They are described in more detail in Chapters 3 and 6. In addition, there are supplementary video lessons in the course on “myCampus”.

The final portfolio is thus the product of the entire editing process with its intermediate steps including the final product, a reflection of the approach, and methodology within the framework of a two-page abstract.

Depending on the module or course, it may be part of the portfolio to procure the resources required for implementation. These can consist of data, applications, technical equipment, software, various tools, etc. Further information on this topic can be found in Chapter 6 of this guide and, if applicable, in the corresponding course on “myCampus”.

Important: The selection, parameterization, and use of programs (software, tools, apps, etc.) are also an integral part of the portfolio for conceptual topics and must be documented accordingly.

The portfolio is submitted via the “PebblePad” tool. For registration, use, and the individual functions of “PebblePad”, there is a separate user guide on “myCampus”. If you have any further questions regarding your exam performance or “PebblePad”, please contact: pruefungsamt@iubh-fernstudium.de.

2 Phases of a portfolio

In order to develop the portfolio, a multi-step procedure is necessary. In the final portfolio, these steps are reflected in various portfolio pages, which cover the phases described below. The focus here, is on the strict adherence to these 3 phases, in the order given and the adequate processing of the respective intermediate steps.
Important: The correct adherence to the given order is part of the examination. If the outlined order is not adhered to, the test will be considered as failed.
In addition, the orientation towards the central guiding questions should serve as further assistance with the development of the portfolio. The portfolio is then created by working as thorough as possible through the individual phases.

2.1 Conception phase

In this phase, the concept or core idea should be introduced contextually with a short description of the content and the goals, as well as the initial motivation. The procedure shall also be described.
Chapter 3 contains the topics and tasks.

The conception phase should be based on the following **key questions**:

- What is the aim of my work?
- What are my ideas? Testing ideas for sustainability: What are positives and negatives?
- What is the concept of my work?
- How do I plan to proceed?
- Which methodology/which tool did I choose and why?

The conception phase concludes with the submission of a first portfolio page in PebblePad in the following form:

- A brief explanation of the concept in the form of a text must be included.
- Depending on the course, first sketches/drafts/diagrams must be created and inserted according to the formalities and parameters defined in chapter 6.

Important: This part must be submitted before the next steps "Development phase/reflection phase" and the "Finalization phase" can be processed. Information on the next steps and phases can be found in chapters 2.2 and 2.3.
After submission, the tutor will provide feedback which will include recommendations. Please proceed to the subsequent phases only after you have received feedback. Whilst proceeding further, the recommended corrections within the scope of the feedback should be taken into consideration as much as possible. Depending on the scope and number of submitted works, feedback can take up to seven days.
The conception phase, together with the other phases, is included in the final grade, see chapter 5.

2.2 Development phase/reflection phase

Implementation of your basic ideas takes place in this phase. Depending on the scope, several intermediate steps may be necessary.

The description of the implementation and the objective of this phase can be defined as follows. The project developed in this course builds on a basic understanding of the Python programming language and aims at understanding important notions and concepts from two domains:

- **Object-oriented programming** such as classes, objects, abstraction, encapsulation, inheritance, polymorphism, composition, and delegation.
- **Functional programming** with ideas such as functions as first-class objects, decorators, pure functions, immutability and higher order functions.

To this end students develop a Python program together with suitable documentation for it, according to the specific goals outlined in chapter 3.
This phase concludes with the submission of a portfolio page in the following form:

- A short, written explanation of the implementation (maximum of one DIN A4 page).
- The creation of a digital design/intermediate step. This should be enclosed according to the formalities and parameters defined in Chapter 6.

Important: This part must be submitted (uploaded and saved) before the "Finalization Phase" can be processed. After submission, the tutor will give some minor feedback. It is recommended further processing of the next phase is continued only after the feedback has been given. The recommended corrections within the scope of the feedback should be taken into consideration as much as possible. Depending on the scope and number of submitted works, feedbacks can take up to seven days.
This phase is included in the final mark, see Chapter 5 for evaluation criteria.

2.3 Finalization phase

In this phase the final product is developed and submitted. A two-page **abstract** must be included in the final concept as well as a reflection on the implementation and its results, e.g. from the following guidelines:

- Short presentation of the **objectives**/the idea/the concept/the methodology
- Sources/**Resources**/Software/Breakdown of implementation, etc.
- From the concept idea to implementation and result(s): What **result** have I achieved? Does the result correspond to the goal of the work? If not, why not?
- If necessary, reflection on your performance, e.g.: What conclusions can I draw from the editing process?

This phase concludes with the development of a third, final portfolio part with the following content:

- An abstract in PDF format. Further specifications and formalities are defined in Chapter 6.
- In addition, the final product should be created and inserted according to the formalities and parameters defined in Chapter 6.
- A link to an One Drive Business folder.

Important: Within six weeks following the submission of the third portfolio page, the tutor submits the final feedback which includes evaluation and scoring. The feedback and the fulfilment of the evaluation criteria can be found in the final portfolio in “PebblePad”. The grade will be visible in the *Transcript of Records* in CARE. If you have any questions about your exam performance, please contact: pruefungsamt@iubh-fernstudium.de

3 Topics and tasks

3.1 Task: Create a Habit Tracking App

Creating good habits and breaking bad ones is no easy task. In order to keep track of certain habits or achieving personal goals, more and more people rely on so called habit trackers to help them throughout the day. If you check out any popular app store, you'll find no shortage of habit tracking applications in a wide range of quality and prices.
Recently, a client has approached you and wants you to help them build a basic Python backend for a habit tracking app, which they want to roll out later this year. To make this project feasible, you must focus on the essential functionality of such an application. You are not asked to provide any sort of graphical user interface, just the basic functionality of a habit tracker using object-oriented and functional programming in Python, according to the following specifications.

Formally, a habit is a *clearly defined task that* must be *completed periodically* (e.g. brush your teeth every day or go to the dentist once every year). The basic building blocks of a tracking app are as follows:

- A user can define multiple habits in the application. A habit has a task specification and a periodicity.
- A *task can be completed*, i.e. "checked-off", by a user at any point in time.
- Each task needs to be checked-off *at least once* during the period the user defined for the respective habit. If a user misses to complete a habit during the specified period, the user is said to *break the habit*.
- If a user manages to complete the task of a habit x consecutive periods in a row, I.e. without breaking the habit, we say that the user established a *streak of x periods*. For instance, if a user wants to work out every day and does so for two full weeks, they establish a 14-day streak of working out.
- The habits users enter in the app are not only stored but can also be analysed. Users want answers to several questions like: what's my longest habit streak? What's the list of my current daily habits? With which habits did I struggle most last month?

Acceptance criteria for your habit tracker

Habit trackers can be complicated pieces of software. The scope of this project is limited due to time-constraints. The following acceptance criteria should help you to get a clearer picture of what's expected of you for this project.

- Your solution should be built using *Python version 3.7 or later,* so that you can focus on writing modern Python code. The tools and libraries used to implement your habit tracker application are entirely your own choice. However, you are not permitted to directly use or modify existing, third-party habit tracking tools that you might find on the internet of elsewhere.
- The software you submit must come with detailed, self-contained *installation and run instructions* so that it is clear how to install the project and use your habit tracker.
- The concept of a habit should be encoded using object-oriented programming.
- Your tracker should be able to let users create two habit periods, namely *weekly and daily* habits.
- Your solution comes with *5 predefined habits* (at least one weekly and one daily habit). It should be clearly documented how new habits can be created with your solution. Also, make transparent how a user can complete a task within a given period.

- For each habit, your system tracks when it has been created, and the date and time the habit tasks have been completed.
- For each predefined habit, you should provide example tracking data for a period of 4 weeks.
- Your solution has an *analytics module* that allows users to analyse their habits. The functionality of this analytics module must be implemented using the *functional programming paradigm*. You are free to consider implementing other functionality as well, but these are the minimal requirements. Provide functionality to
  - return a list of all currently tracked habits,
  - return a list of all habits with the same periodicity,
  - return the longest run streak of all defined habits,
  - and return the longest run streak for a given habit.
- Your solution has to have a clean API that users understand. You do this by exposing a command line interface (CLI) tool to the user that allows them to create, delete and analyse their habits.
- The critical parts of your solution, in particular the validity of your habit tracking components and the analytics module, should be tested by providing a *unit test suite* that can be run following the instructions provided with the solution.

Your application needs to be built, documented and delivered according to the following three phases.

3.1.1 Conception phase

This phase represents the most important part of the design process of your project. Anything that is overlooked or forgotten in this phase has a negative effect on the implementation later and might lead to a failed project.

The first step is to create a written concept, to describe everything you need to build your habit tracking application. You will create at least one diagram (e.g. in UML) and put it into the written concept to show the interaction of the components and the process. This step is perhaps the most important of the entire design process. Sufficient time is needed before the next steps can be taken. It is therefore essential to follow the sequence of the respective steps carefully.

It is important not only to consider what the client probably expects from your habit tracking app, but also how users should interact with it, how they create new habits, complete tasks and check for their progress. The written concept must explain why the structure and process have been designed in the particular manner.

Think about which tools you can use to implement each component and the communication between the components. A conceptual text (1 DIN A4 page) must be prepared for the submission, explaining these analyses and considerations, together with your diagram(s), showing the interaction of the components and the process.

Throughout the process of this phase, you can attend online tutorials. They provide an opportunity to talk about ideas and drafts or obtain feedback. Everyone can get involved and learn from each other's feedback in tutorials. You can also access other channels to address questions to your tutor and your fellow students. It is recommended to make use of these channels to avoid common mistakes. You only submit work after making use of the above-mentioned tutorial and informative media. This will be followed by a feedback from the tutor and the second phase work can begin.

3.1.2 Development/reflection phase

In this phase you start implementing your habit tracking app:

- You set up the frameworks and tools that you described in the conception phase.
- You implement the components outlined in your design diagram.
- Your code is commented, and you document its usage.
- Users can create, manage and inspect habits they define in a convenient way.
- Users can analyse their habits.

In this phase you must submit an explanation of your design and implementation procedure as a composite presentation PDF with about 5-10 slides. The file should contain visual elements that facilitate comprehension, it needs to be structured and include hyperlinks to the tools you decided to use.

Throughout the process, online tutorials and other channels provide you with the opportunity to discuss ideas and get sufficient feedback, tips, and hints. It is recommended to use these channels to avoid errors and to improve your work. Once this is done, you can hand in your second phase for evaluation. Following feedback from the tutor, your work on the final draft will continue in the third phase.

3.1.3 Finalization phase

In this final phase, your goal is to polish and refine the application, after having received feedback from the tutor, and prepare it for final submission. Certain elements may have to be improved or changed to finalize the task and complete this portfolio course.

Your finished product, the habit tracking app, is submitted by providing all files: the files you created, your Python program code, documentation etc., as follows:

- Your project is hosted on a public GitHub repository. That means you must create a GitHub account, if you don’t have one already, create a repository and upload all your code into this repository. On submission in PebblePad you provide a link to the repository. Hosting the project on GitHub is part of building your portfolio.
- You then create a ZIP file from all files contained in the GitHub repository and put it into an OneDrive Business folder. You will provide a link to this folder in your submission in PebblePad.

Important note: The content of the GitHub repository and the contents of the ZIP file uploaded to your OneDrive Business folder should be *identical*. You are not allowed to modify either one after the final submission. The content of the OneDrive folder will be sent to the examination office upon submission.

Additionally, you provide a 2-page abstract PDF document in which you describe your solution in terms of content and concept. This abstract presents a short break-down (“making of” of the project) about the technical approach in a clear and informative way. To summarize, in this phase you will submit your project by hosting it on GitHub, uploading it to OneDrive Business as a ZIP file and provide the link in PebblePad as well as an abstract explaining the project and the final product (scripts with installation manual and documentation).

In the “Finalization phase”, the online tutorials and other channels provide you with the opportunity to obtain sufficient feedback, tips, and hints before the finished product is finally handed in. It is recommended to use these channels to avoid errors and to make improvements. The finished product is submitted together with all the above materials. Following the final submission, the tutor submits the final feedback which includes evaluation and scoring within six weeks.

4 Tutorial support

In principle, several channels are open to attain feedback for the portfolios. The respective use is the sole responsibility of the user. The independent development of a product and the work on the respective portfolio parts is part of the examination performance and is included in the overall assessment.

On the one hand, the tutorial support provides feedback loops on the portfolio parts to be submitted in the context of the conception phase as well as the development and reflection phase. The feedback takes place within the framework of a submission of the respective part of the portfolio. In addition, regular online tutorials are offered. These provide you with an opportunity to ask any questions regarding the processing of the portfolio and to discuss other issues with the tutor. In the course on “myCampus” there is also a forum available to clarify course-specific questions with all course participants and the tutor. The tutor is also available by email for technical consultations as well as for formal and general questions regarding the procedure for portfolio management.

Technical questions regarding the use of “PebblePad” should be directed to pruefungsamt@iubh-fernstudium.de

5 Evaluation

The following criteria are used to evaluate the portfolio with the percentage indicated in each case:

Problem Solving Techniques	*Capturing the problem *Clear problem definition/objective *Understandable concept	10%
Methodology/Ideas/Procedure	*Appropriate transfer of theories/models *Clear information about the chosen methodology/idea/ procedure	20%
Quality of implementation	*Quality of implementation and documentation	40%
Creativity/Righteousness	*Creativity of the solution approach *Solution implemented fulfils intended objective	20%
Formal requirements	*Compliance with formal requirements	10%

The design and construction of the portfolio should consider the above evaluation criteria, including the following explanations.

Problem Solving Techniques: According to the topic a habit tracking app should be designed, implemented and documented. It must be clear how to set up, run and use the application using the transferred files.

Methodology/idea/procedure: The concept of habits and how to track them according to their tasks and periods should follow a sound underlying object-oriented design. Analysing, summarizing and aggregating habits over time should be suitably implemented using functional programming techniques. Which design and technology choices drive the project? Are the choices justified and do they make sense within the scope of this portfolio course?

Quality of implementation*:* Does the implementation follows a clear concept? Can it easily be used by a user that is unfamiliar with the project, e.g. by providing sufficient documentation? Is it straightforward to create and track new habits with the provided solution? Is it easy for users to analyse their habits with the tool? The acceptance criteria laid out in Chapter 3 must be respected.

Creativity/Rightness: It is evaluated whether the specific requirements have been understood and implemented in a comprehensible and innovative way. The basic functionality of the application must be covered by unit tests to ensure correctness of the software.

Formal requirements: The submission follows the acceptance criteria from Chapter 3 and the formal guidelines following in the next chapter. It is particularly important to respect the formal submission requirements outlined in Chapter 6.

6 Formal guidelines and specifications for submission

6.1 Components of the examination performance

The following is an overview of the examination performance portfolio with its individual phases, individual performances to be submitted, and feedback stages at one glance. A template in “PebblePad” is provided for the development of the portfolio parts within the scope of the examination performance. The presentation is part of this examination.

Stage	Intermediate result	Performance to be submitted
Conception phase	Portfolio part 1	– One page concept in written form, additionally at least one diagram to show interaction and processes.
	Feedback	
Development phase/ reflection phase	Portfolio part 2	- Explanation of implementation in written form as a composite presentation PDF with about 10 slides; incl. visual elements & hyperlinks.
	feedback	
Finalization phase	Portfolio part 3	- -2 page abstract (making of) - Final product (scripts with installation manual and documentation included) - Linked OneDrive business folder (incl. all files)
	Feedback + Grade	

6.2 Format for Digital File Submission

Conception phase	
Recommended tools/software for processing	– Word or LaTex – Tools to draw software models (e.g. Visio, SmartDraw, Edraw) – Relevant sources in digital forms
Permitted file formats	PDF
File size	Max. 15 MB
Further formalities and parameters	Files must always be named according to the following pattern: For the performance-relevant submissions on “PebblePad”: Name-FirstName_MatrNo_Course_Topic_Submission_Concept.pdf Example: Mustermann-Max_12345678_OOFPP_Habits_Submission_Concept.pdf For feedback between submissions-by email: Name-FirstName_ MatrNo_Course_Topic_Feedback_Concept.pdf

	Example: Mustermann-Max _12345678_OOFPP_Habits_Feedback_Concept.pdf

Development / reflection phase	
Recommended tools/software for processing	– Powerpoint or LaTex – Researched tool selection for task implementation – Relevant sources in digital forms
Permitted file formats	PDF
File size	Max. 15 MB
Further formalities and parameters	For the performance-relevant submissions on “PebblePad”: Name-FirstName_MatrNo_Course_Topic_Submission_Concept.pdf Example: Mustermann-Max_12345678_OOFPP_Habits_Submission_Concept.pdf For feedback between submissions-by email: Name-FirstName_ MatrNo_Course_Topic_Feedback_Concept.pdf Example: Mustermann-Max _12345678_OOFPP_Habits_Feedback_Concept.pdf

Finalisation phase	
Recommended tools/software for processing	– Powerpoint or LaTex – Researched tool selection for task implementation – Relevant sources in digital forms
Permitted file formats	– PebblePad: PDF – **GitHub:** various file formats for your solution implemented in Python, helper code, e.g. Shell scripts, etc., and other related files like documentation, installation instructions, etc. – OneDrive: various file formats for submissions from all three phases. The code submitted via GitHub also has to be uploaded to OneDrive by creating a ZIP file from all the files.
File size	– PebblePad: Max. 25MB – OneDrive: Max. 150 MB
Further formalities and parameters	**IMPORTANT:** you need to provide a link to the folder on your “OneDrive Business” that has been created especially for the submission (please follow the instructions on “myCampus”). This folder contains all the files you used to complete the task. To ensure a better overview, please create subdirectories for this purpose. The folder structure then looks like this:

	•Main directory (this is linked) -> name:Name-FirstName_MatrNo_AIUseCase_Topic ○ Subdirectory -> name: OOFPP_Topic_P1 ○ Subdirectory -> name: OOFPP_Topic_P2 ○ Subdirectory ->name: OOFPP_Topic_P3 In phase 3 you should also upload a zipped version (ZIP file with all program code) to OneDrive with the following naming convention: Name-FirstName_MatrNo_Course_Topic_Submission_Code.zip Example: Mustermann-Max_12345678_OOFPP_Habits_Submission_Code.zip For the submission of your code on GitHub you are free to choose your own username. You create a public GitHub repository of your choice and upload all your code to it. That means your code should be available under the following URL schema: https://github.com/<user_name>/<repository_name> Example: https://github.com/maxmustermann/oofpp_habits_project For the performance-relevant submissions on "PebblePad", The 2-page abstract (making of) must always be named according to the following pattern: Name-FirstName_MatrNo_Course_Topic_Submission_Abstract.pdf Example: Mustermann-Max_12345678_OOFPP_Habits_Submission_ Abstract.pdf

6.3 Format of Abstract

Length	2 pages of text
Paper size	DinA4
Margins	Top and bottom 2cm; left 2cm; right 2cm
Font	General Text - Arial 11 pt.; Headings - 12 pt., Justify
Line Spacing	1.5
Sentences	Justified; hyphenation
Footnotes	Arial 10 pt., Justify
Paragraphs	According to mental structure - 6 pt. after line break
Affidavit	The affidavit shall be made in electronic form via "myCampus". No submission of the examination performance is possible before it. Please follow the instructions for submitting a portfolio to "myCampus".

6.4 Quotation and Footnotes

It is absolutely forbidden to use ideas, statements and/or facts that are not the writer's own, without attributing the source. This is plagiarism. Citations should follow the American Psychological Association (APA) style. For direct quotes used in the text, use the following citation format: author's last name, year of publication, and page number, all in parentheses (name, year, page). For example:

> *The role of Mephistopheles can be described as follows: "Man loves peace, and for this reason he needs an opponent of the same strength, that can motivate him in the face of life's perpetual difficulties" (Schmidt, 2004, p. 102). This makes clear that…*

Indirect quotes do not need page numbers but do still need to include the author's last name and year of publication. While footnotes are not recommended by the APA style guide, if used they should be placed at the end of the corresponding page, and marked within the text by a superscript number[1].

6.5 Submission

The portfolio has to be submitted via the PebblePad portal. How to use PebblePad can be found in a separate manual on myCampus.

As with other assessment methods, an affidavit is also required for the portfolio. This is done electronically via myCampus.

If you have any questions about PebblePad, please contact pruefungsamt@iubh-fernstudium.de

Good luck creating your portfolio!

[1] This is an example of how footnotes can be used to mark border annotations.