**Vidzeme University of Applied Sciences**

**Faculty of Engineering**

Introduction to Python and Data Exploration

**group 06**

**practical work #1**

Valmiera, 2024

**Table of Contents**

[1.0 Content 4](#_Toc159935023)

[2.0 Tasks and their solutions 5](#_Toc159935024)

[2.1 Task 1 5](#_Toc159935025)

[2.2 Task 2 5](#_Toc159935026)

[2.3 Task 3 6](#_Toc159935027)

[3.0 Chapter 2 6](#_Toc159935028)

[4.0 References 10](#_Toc159935029)

[5.0 Appendix (Code) 11](#_Toc159935030)

[5.1 Task\_1.1\_Gr.06.py 11](#_Toc159935031)

[5.2 Task\_1.2\_Gr.06.py 11](#_Toc159935032)

[5.3 Task\_1.3\_Gr.06.py 11](#_Toc159935033)

|  |  |  |  |
| --- | --- | --- | --- |
| Document versions | | | |
| Version | Status / Changes | Date | Author |
| 0.1 | First version implementation | 02. 27.2024 | Emīls Konrāds |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Contacts and responsible (-s) | | | |
| Name Surname | Department | Position | Contact information (e-mail) |
| Emīls Konrāds | Group #06 | Coordinator | emils.konrads@va.lv |
| Gatis Jurisons | Group #06 | Member | gatis.jurisons@va.lv |
| Kristers Kalniņš | Group #06 | Member | kristers.kalnins@va.lv |
|  |  |  |  |
|  |  |  |  |

# Content

**Goal:** The team (further referenced as team 06) must complete the given task by the client (university professor) described in task 1 of unit 2.

**Content**: two charts, 597 words, six images, three appendixes.

# Tasks and their solutions

Team 06 organized a meeting after the given the task, on the Monday of the next week and started organizing a workflow and work management chain. The team 06 decided to split the workload in 2 different categories:

* Documentation
* Coding

While all participants take part in all parts of the workload, some team members take more of a lead in documenting and only help in the coding process. For the work flow a GitHub repository was created and maintained by the team coordinator. By the end of each week each team participant must send in a flow chain of how they see the app working, each demonstrating their plan of development.

## Task 1

The First task is a basic calculator that multiplies hours with rate and displays a salary in the command port. (Osis, Moodle.va.lv, 2024)

A screen shot of a computer code

Description automatically generated

Image: 1.1. Code from (Task\_1.1\_Gr.06.py)

## Task 2

The second task was to add an overtime calculation to the application. Witch was added with an if else statement and simple additions to the already created code. The main point is where we check if the weekly hourly rate exceeds 40 hours to check if the worker has worked more than the minimum given hour rate. (Osis, Moodle.va.lv, 2024)

A screen shot of a computer program

Description automatically generated

Image: 1.2. Code from (Task\_1.2\_Gr.06.py)

## Task 3

The final task was to “dummy proof” the application by making some simple try catches if a user would add a not matching input to the calculation. (Osis, Moodle.va.lv, 2024)

A computer screen shot of text

Description automatically generated

Image: 1.2. Code from (Task\_1.3\_Gr.06.py)

# Chapter 2

As described in the starting chapter before all the participants of group 06 had handed in Flowcharts of how each one of the participants sees the app working. This was all managed using discord where each participant sent their work in for review.

A diagram of a flowchart

Description automatically generated

Image: 2.1. Gatus Jurisons Flowchart

A diagram of a work flow

Description automatically generated

Image: 2.2. Kristera Kalniņa Flowchart

A diagram of a program

Description automatically generated

Image: 2.3. Emīla Konrāda Flowchart

# References

Osis, K. (2024, Feb). *Moodle.va.lv.* Retrieved from 2024\_practical\_work\_1.pdf: https://moodle.va.lv/pluginfile.php/31044/mod\_resource/content/7/2024\_practical\_work\_1.pdf

Osis, K. (2024, Feb). *Moodle.va.lv.* Retrieved from Template of Document: https://moodle.va.lv/mod/resource/view.php?id=11925

# Appendix (Code)

## Task\_1.1\_Gr.06.py

hours = float(input("Enter Hours: "))

rate = float(input("Enter Rate: "))

pay = hours \* rate

print("Your salary is:", pay, "EUR")

## Task\_1.2\_Gr.06.py

hours = float(input("Enter Hours: "))

rate = float(input("Enter Rate: "))

if hours <= 40:

    pay = hours \* rate

    print("Your salary is:", pay, "EUR")

else:

    pay = 40 \* rate + (hours - 40) \* rate \* 1.25

    print("Your salary is:", pay, "EUR")

## Task\_1.3\_Gr.06.py

hours = float(input("Enter Hours: "))

rate = float(input("Enter Rate: "))

if hours <= 40:

    pay = hours \* rate

    print("Your salary is:", pay, "EUR")

else:

    pay = 40 \* rate + (hours - 40) \* rate \* 1.25

    print("Your salary is:", pay, "EUR")