Leszek Ciopiński, PhD
2023/2024

# Fundamentals of Programing 2

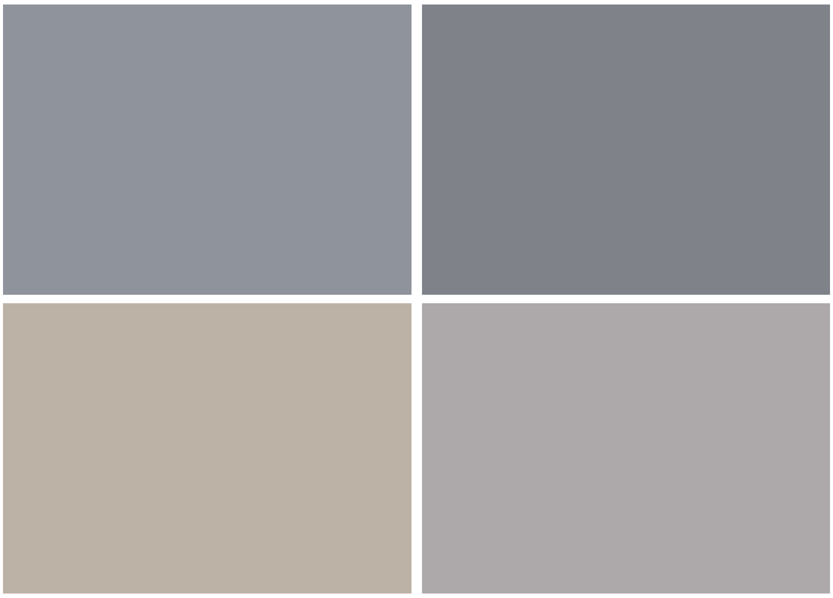
## Project

## 1. Introduction

In this document, the rules and terms of the assessment of a project are presented. It is also published on Achilles portal (https://achilles.tu.kielce.pl).

"Algorithms + Data Structures = Programs"

*-Niklaus Wirth*



## 2. Contact

- Stationary, during the classes or consultation hours.
- Via WebEx, during the consultations. The schedule of consultations and link to a WebEx room are published on the Achilles portal.
- Via email: l.ciopinski@tu.kielce.pl

## 3. Teams

- All students are split into working groups (depending of number of students, into two-person and/or three-person groups).
- Information about  members of each group and the chosen project topic should be sent to the teaching assistant  via email to: l.ciopinski@tu.kielce.pl by the end of March at latest. Each group receives a reply about accepting or rejecting the topic. Each topic could be chosen by only one group.
- Failure to submit requested information before the deadline will result in negative note for the course.

## 4. Project Results

- A program written in C language (neither C++ nor C#) together with its source code
- The project report, which should contain:
  - Topic and its number

    - An abstract - very short (a few sentences) description of the project
    - Information, what has been accomplished.
    - Information, what has not been accomplished and why.
    - Bibliography (also links to websites)
- additional, items which are necessary to run the project (if applicable)

## 5. Evaluation of a project

- The finished project should be sent to the teaching assistant at least 4 days before the last classes.
- The Project evaluation will be based on:
    - program features (its accordance with the topic)
    - a source code quality
    - a performance and a stability of the program

## 7. Topics

1. Travelling Salesman Problem
   Finding a solution using The nearest neighbour algorithm. A list of cities and distance between them should be read from a file.
   Three-person group: Cost of transport from city A to B is different than from city B to A.

2. Equation Calculator
   A calculator which input is an equation. The equation contains digits, brackets "( )" and operators "+, -, *, /, ^(power)". TIP: Use Reverse Polish notation (RPN) to solve a problem with brackets.
   Two-person group could omit operators: „/” and „^”.
   https://en.wikipedia.org/wiki/Reverse_Polish_notation

3. Vigenère cipher
   Write a program that could encrypts and decrypts a text file using Vigenère cipher.
   Three-person group should service a file which contains uppercase, lowercase, digits and special characters, like space, !@#$..

4. An escape from a labyrinth
   Using the A* algorithm, find the shortest way from a selected place in labyrinth to the

exit.
Three-person group: Store the escape path with map to a file.

5. Homophonic Substitution Cipher
Use Homophonic Substitution cipher to encrypt and decrypt a text file. A three-person group should implement an algorithm which uses uppercase and lowercase letters and special characters, like space, !@#$.
https://en.wikipedia.org/wiki/Substitution_cipher#Homophonic_substitution

6. Book cipher
Write a program that could encrypts and decrypts a text file using Book cipher. Three-person group should service a file which contains uppercase, lowercase, digits and special characters, like space, !@#$.
https://en.wikipedia.org/wiki/Book_cipher

7. Address Book
Write a program that collects contact data (eg. name, surname, phone number, addresses etc.). Accessing to the program should equire a user id and password. The program should allow the user to edit the collected data (add, modify or delete) and search information.
Three-person group should allow to search information according to given pattern, eg. A* = Adam, Ann ('*' means any substring, '?' means any character)

8. Library System
Write a program that collects data about books and its borrower (eg. user name and surname, an author and title of a book, time to return). Accessing to the program should equire a librarian id and password. The program should allow the user to edit the collected data (add, modify or delete) and search information.
Three-person group should allow to search information according to given pattern, eg. A* = Atlas, Almanac ('*' means any substring, '?' means any character)

9. System for Car Rental
Write a program that collects data about cars and its users (eg. user name and surname, a car brand, time to return, cost, etc.). Accessing to the program should equire a staff member id and password. The program should allow the user to edit the collected data (add, modify or delete) and search information.
Three-person group shoud allow to search information according to given pattern, eg. P* = Polonez, Panda ('*' means any substring, '?' means any character)

10. Warehouse System
Write a program that collects data about goods and their quantity in a warehouse. Accessing to the program should equire a staff member id and password. The program should allow the user to edit the collected data (add, modify or delete) and search information.
Three-person group shoud allow to search information according to given pattern, eg. w* = wheel, workout equipment ('*' means any substring, '?' means any character)

11. Ticket Selling Support System
    Write a program that collects data about tickets and their quantity. Accessing to the program should equire a staff member id and password. The program should allow the user to edit the collected data (add buyers, modify owner or delete - ticket return) and search information.
    Three-person group shoud allow to manage more than one ticket.

12. System for Bike rental
    Write a program that collects data about bikes and its users (eg. biker name and surname, a bike brand, a bike type, time to return, cost, etc.). Accessing to the program should rquire a staff member login and password. The program should allow the user to edit the collected data (add, modify or delete) and search information.
    Three-person group shoud allow to search information according to given pattern, eg. *s = Kross, NS Bikes ('*' means any substring, '?' means any character)

13.

14.

15.

16.

17.

18.

19.

20. "OPEN"
    Any topics proposed by students and accepted by the teaching assistant.