# CP4P Final Project

## Introduction

CP4P students will create a demo application to demonstrate various operations with null-terminated C strings.

## Project modules

Module #1 – Strings Fundamentals (fundamentals.h, fundamentals.c)

Module #2 – Strings Manipulations (manipulating.h, manipulating.c)

Module #3 – Strings Conversions (converting.h, converting.c)

Module #4 – Strings Tokenizing (tokenizing.h, tokenizing.c)

Each team has three “junior” programmers responsible for modules 1-3 and one “senior” programmer (team leader) responsible for module 4, synchronizing and integrating modules into the main application.

## Projects Stages and Deliverables (one week for stages 1-3 approx.)

Stage #0 – preparing and testing the necessary tools: git command line utility, open source g++ compiler, and MS Teams. Open source g++ complier is strongly recommended, but Visual Studio and Xcode could be also used.

Stage #1 – submitting version #1

Stage #2 – submitting version #2

Stage #3 – submitting final version #3

## Project Details

Students will not develop modules (we will not duplicate or interfere with IPC144). All the modules versions will be provided by the teacher as .png (graphics) files. So, students will have to ***type*** code, ***comment***, ***compile***, ***test***, ***stage*** and ***commit*** versions using git. They will ***communicate*** and ***collaborate*** through MS Teams.

## Grading Approach

We may mark assignment with C/C+ if stage #1 was completed, B/B+ if stage #2 was completed, and A/A+ if students reached the final stage #3. Final mark will also depend on the quality of students’ comments, efficiency of their communications monitored by the teacher through MS Teams, ability to meet deadlines and, of course, on final application testing results. Matrix will not be used for submissions. The team leader is responsible to send versions (1 to 3) source code and testing screenshots to the teacher by email.

## Appendix A C String Functions used by Modules

### Fundamentals Module

strlen() // length

strcpy() // copy

### Manipulating Module

strcat() // concatenation

strcmp() // comparison

strstr() // search

### Converting Module

atoi() // string to int

atof() // string to double

atol() // string to long

### Tokenizing Module

strtok() // tokenizing

## Appendix B Students’ Responsibilities, Marks, and Tools Used

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Version 1**  **Grade: “C/C++”**  **Tools: g++** | **Version 2**  **Grade: “B/B++”**  **Tools: g++, git** | **Version 3**  **Grade: “A/A++”**  **Tools: g++, git** |
| **Junior Programmer #1**  **(Fundamentals)** | Indexing | Add Measuring | Add Copying |
| **Junior Programmer #2**  **(Manipulating)** | Concatenating | Add Comparing | Add Search |
| **Junior Programmer #3**  **(Converting)** | Converting to int | Add Converting to  double | Add Converting to long |
| **Senior Programmer/Leader**  **(Tokenizing)** | Tokenizing words | Add Tokenizing phrases | Add tokenizing  sentences |

## Appendix C Deliverables and Deadlines

### Version 1 “C/C+” grade (week 1)

1. fundamentals.h
2. fundamentals.c
3. manipulating.h
4. manipulating.c
5. converting.h
6. converting.c
7. tokenizing.h
8. tokenizing.c
9. main.c
10. test\_screenshot.txt

### Version 2 “B/B+” grade (week 2)

1. fundamentals.c
2. git\_status\_log1\_screenshot.txt
3. manipulating.c
4. git\_ status\_log2\_screenshot.txt
5. converting.c
6. git\_ status\_log3\_screenshot.txt
7. tokenizing.c
8. git\_ status\_log4\_screenshot.txt
9. main.c
10. test\_screenshot.txt

### Version 3 “A/A+” grade (week 3)

1. fundamentals.c
2. git\_ status\_log1\_screenshot.txt
3. manipulating.c
4. git\_ status\_log2\_screenshot.txt
5. converting.c
6. git\_ status\_log3\_screenshot.txt
7. tokenizing.c
8. git\_ status\_log4\_screenshot.txt
9. main.c
10. test\_screenshot.txt