This document describes the **team work assignment** for UBT master students studying Advanced Programming Course.

Project Assignment

Implement an application in an innovative field by choice. Here are some alternatives:

- An **OOP application**, which will include the following object-oriented assets:
    - At least **5 interfaces/abstract class** (with one or more implementations)
    - At least **15 classes** (implementing the application logic)
    - At least **1 exception class** (with usage in your code)
    - At least **3 levels of depth in inheritance**
    - At least **1 polymorphism** usage
    - At least **1 enumeration**
    - At least **1 architectural style**
    - At least **3 design patterns** (e.g. Composite, Singleton, Factory, Wrapper, Bridge, Command, Iterator, …)

- A **big data, grid computing, blockchain or peer-to-peer application**
    - **Peer-to-Peer** networking and applications disseminates state-of-the-art research and development results to facilitate effective deployment of P2P networking and applications
    - A **grid computing** system is a type of parallel computing system which enables us to share computing power, disk storage, databases and software apps. The core functional computational requirements for grid **applications** are:
        - The ability to allow for independent management of computing resources
        - The ability to provide mechanisms that can intelligently and transparently select computing resources capable of running a user's job
        - The understanding of the current and predicted loads on grid resources, resource availability, dynamic resource configuration, and provisioning
        - Failure detection and failover mechanisms
        - Ensure appropriate security mechanisms for secure resource management, access, and integrity.
        - E.g., SETI@home, World Community Grid
    - Implementation of a **blockchain application**. Some ideas described here
    - **Big data** is a field that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional data-processing application software.
- A **parallel processing application**
    - Compare the performance of different parallel algorithms for the same task on different machines (often different algorithms are best for different platforms)
    - Add high-performance parallel bindings to a system like Javascript or Python. e.g., See PyCuda and the great projects in 2014.
    - Any parallel computing project idea described here
- Any **other project** related to course topics discussed during classes

---

E. Jajaga & A. Kadriu

General Requirements

Your final project gives you the opportunity to dive deeply into advanced programming. What you attempt for your project is completely up to you. However, you are expected to show basic and advanced knowledge on object-oriented programming or any other emerging programming paradigms. You are recommended to divide modules to each member of the team, while all of you should understand the entire code. You can use open source applications and provide additional changes/improvements to it (strongly recommended). Moreover, you should do a research about similar approaches dealing with your hypothesis.

Additional Requirements

- Follow the **best practices for OO design**: use data encapsulation, use exception handling properly, use inheritance, abstraction and polymorphism properly, and follow the principles of strong cohesion and loose coupling.
- Obligatory use any versioning system such as **Git** (https://github.com/) to keep your source code and for team collaboration.
- Optionally, provide a **class diagram** (to visualize all types).

Non-Required Work

Completely finished project is not obligatory required. It will not be a big problem if your project is not completely finished or is not working greatly. This team work project is for educational purpose. Its main purpose is to experience **advanced programming skills** in a real-world project and to get some experience in **team working** and team collaboration with Git.

Deliverables

Put the following in a **ZIP archive** and submit it (a representative team member submits the file):

- **Project Proposal**: The proposal should consist of 1-2 pages describing the team members and the problem you plan to solve, outlining how you plan to solve it, and describing what you will "deliver" for the final project. We will discuss with every group before the project proposal to help you refine your topic and would be happy to provide feedback on a draft of your proposal before it is due.
- The complete **source code**.
- **Report** of your project in the form of a **conference paper**/poster/demo (follow the template format). It should provide the following information (in brief):
    - Application Name
    - Team name and list of team members
    - Project purpose – what problem do you solve?
    - Class diagram of your types (optionally)
    - Describe the advantages of your approach versus the others (Related Works)
    - Describe important fragments of codes
    - Screenshots of your application's UI
    - The URL of your Git repository (do not remove the repo until publishing the results)
    - Any other information (optionally)
    - Follow the template published in Moodle
- Optionally provide a **PowerPoint presentation** designed for the project defense.

Public Project Defense

Each team will have to deliver a **public defense** of its work in front of the other students. You will have **only 20 minutes** for the following:

- **Demonstrate** the application (very shortly).
- Show the **class diagram** (just a glance).
- Show the **source code** in the **Git** web-based source code browser.
- Show the **commits logs** to confirm those team members who have contributed.
- Optionally you might prepare a PowerPoint presentation (3-4 slides).

Please be **strict in timing**! Be **well prepared** for presenting maximum of your work for minimum time. Bring your own laptop. Test it preliminary with the multimedia projector. Open the project assets beforehand to save time. You have **20 minutes**, no more.

Give Feedback about Your Teammates

You will be invited to **provide feedback** about all your teammates, their attitude to this project, their technical skills, their team working skills, their contribution to the project, etc. The feedback is important part of the project evaluation so **take it seriously** and be honest.