Rules

Rules - General

- Questions are asked through Zoom. Click on “Ask for Help” and we will help you as soon as possible.
- The final submission of your code should follow good C++ practicces.
- You are to sit in an undisturbed environment without any other people in the same room. You should be visible and connected to Zoom at all times, except during the planned breaks.
- Du will be identified during the exam. Have your photo ID ready.
- Failure to follw the rules will result in a failing grade.
- Part I and Part II must be submitted no later than 11:30.
- Part III will be available from 11:45.
- 11:30 to 11:45 will be a planned break.
- Part III has deadline 13:00.

Rules - aids

- All forms of communication is forbidden, except with the course personnel.
- All forms of copying is forbidden.
- Every source you use for inspiration should be cited.
- Your solutions is submitted according to the instructions of that assignment.
- Be ready to demonstrate your answers after the exam.

Rules - Grading

The exam consists of three parts. Complete solutions/answers to part I and part II is required for a passing grade. It is also required that you have submitted to the “Examination rules” submission in Lisam, which confirms that you swear to follow the rules. You have 3 hours to complete part I and part II. Plan your time accordingly.

The third part is designated for higher grades. It will be published after 3 hours have passed. You have until the end of the exam to complete the assignment(s) in part III.

Part III consists of two assignments.

- To get VG you need to solve *one* assignment and answer each of the related questions.

Agree to the examination rules

Before starting to work on Part I you must submitt the message **“I have read and understood the rules of the examination, and I swear to follow those rules”** to the submission called “2020-05-23: Examination rules (08:00 - 14:00)” in Lisam (see below).

Do this before starting the exam!

Part I

Assessment of part I

For a passing grade on this part you must:

- follow all instructions and requirements presented in the assignment
- make sure that your code follows good programming practices
- write classes that have a clear responsibility and function that has a well defined purpose
- have good encapsulation and resource management

Instructions for submitting part I

You submit your solution through Lisam. You can find the the submissions page here: `https://studentsubmissions.app.cloud.it.liu.se/Courses/TDDE18_2019HT_7K/admin/opportunities`. You can also find it by going to the course page on `http://lisam.liu.se` and going to the “Submission” in the menu. There you should see the following submissions (note that these will not be visible until the start of the exam):

- 2020-05-23: Examination rules (08:00 - 14:00)
- 2020-05-23: Part I (08:30 - 11:30)
- 2020-05-23: Part II (08:30 - 11:30)
- 2020-05-23: Part III (11:45 - 14:00)

Attach the files you want to submit. Hold `Ctrl` to select multiple files. Confirm the submission. You should get a confirmation E-mail.

Your submission must be well tested and must compile with `g++` version 7 with the following flags: `-std=c++11 -Wall -Wextra -Wpedantic -Weffc++` on Ubuntu 18. You can test this by using ThinLinc.

Part I - Assignment (08:30 - 11:30)

Start by confirming that you have read and accept the rules by making a submission to *"2020-05-23 Examination rules"*. If you don't do this your exam will automatically fail.

You must create a program consisting of at least two classes. Both classes should be closely related to the theme TEMA. Your program must demonstrate your knowledge of *constructors*, *data member*, *member functions* and *encapsulation*. You must also create a main program that demonstrates how your classes are used by testing each part of each class.

- One of the classes is *allowed* to be a simple aggregate.
- The other class must demonstrate *at least* two examples within **each** of the specified subjects (*constructors*, *data member*, *member functions* and *encapsulation*). At least one of the member functions must take parameters and perform a operation relevant to the class. Only returning a data member is too trivial.

It should also be possible to print one of your classes with `operator<<`.

Remember to always use the specified theme TEMA. [1] In order to help you with your creativity here are some suggested words/phrases related to TEMA, that might help you figure out what your classes should be and do:

- PHRASE

Remember that it is up to you to show how much knowledge you have and that you have a lot of freedom to actually show your knowledge.

[1] Observe that you don't have to be an expert within the given theme. If you feel like you don't have any knowledge within this theme, then it is fine to make up your own facts.

Part II

Assessment part II

For a passing grade on this part you must:

- follow all instructions and requirements presented in the assignment
- correctly describe how your classes work
- describe how you arrived at your solution

Instructions for submitting part II

This assignment should be answered with text. You need to use a program where you can write headers, text and code examples. You could use Microsoft Word, OpenOffice or LibreOffice. It is also OK to use a purely textual format (for example markdown). The important part is that there is a clear distinction between headers, text and code. You must also be able to export your answers as a PDF. **This part must be possible to read without first reading your solution to part I**.

You must write somewhere between 500 and 2000 words. There are plenty of ways to calculate the word count of a document. Check how you can do it in your program, or use an online application.

You must submit your document as a PDF (one file) to the submission called “2020-05-23: Part II (08:30 - 11:30)” in Lisam (which will submit it to Urkund).

Part II - Assignment (08:30-11:30)

In this part you will explain the code you wrote for part I. Below there is a list everything that must be included in your answer. You could for example create one header for each item. You must include ***ALL*** of these in your answer. We are not looking for “right” or “wrong” answers here. Instead we want to understand your thought process and how you think.

1. Describe how you came up with your solution. Here we are looking for how you reason.
2. Describe how your code demonstrates your understanding of *data member* and explain how they are used in your code.
3. Describe how your code demonstrates your understanding of *constructors* and explain how they are used in your code.
4. Describe how your code demonstrates your understanding of *member functions* and explain how they are used in your code.
5. Describe how your code demonstrates your understanding of *encapsulation* and explain how it is used in your code.
6. Describe how your code demonstrates your understanding of *operator overloading* and discuss how your implementation of `operator<<` works.

Remember to demonstrate all code and why you added it if you want that piece of code to be included in the assessment.