**Exercise work**

Exercise work covers 4 weekly programming exercises (= 40 points). The workload (= amount of hours spent on the exercise work shall be 4 times the hours spent on weekly exercises).

Preferably exercise work is done in groups of 2-4 people.

**Team up with similar minded people:**

1. I just want to pass this course, grade 1 from exercise work is ok for me. I’ll be happy if I can put a bit more effort and maybe get even grade 2 from exercise work.
2. I can and want to put a bit more effort into exercise work and I’m happy with grades 2-3 from exercise work.
3. I want to learn as much as possible and I’m willing to put a lot of time and effort into the exercise work so that I will achieve grade 4, even grade 5 from exercise work.
4. I want to learn as much as possible and I’m willing to put a lot of time and effort (= as much it needs) into the exercise work so that I will achieve grade 5 from exercise work. I will choose the topic so that it challenges me and my skills. I’m willing to even put a bit more effort into the work than needed to achieve an excellent grade.
5. I just want to have a high grade no matter what, but to be honest I’m not willing to put effort in learning and I don’t even care about the learning so much, I only care about (high) grades from exercise work.

**General instructions:**

1. **Go through** all your returned weekly programming exercises (exercises 1 – 8).
   1. Go through every exercise and list every task from every exercise you have not finished.
   2. Especially if you aim at high grade (4 – 5), add as many additional features to your exercise work as you have unfinished tasks in your exercises 1 – 8, topics of those features shall cover the topics of the tasks.
2. **Estimate**, how many hours you have spent on this course
   1. doing the weekly exercises
   2. and keep track of how many hours you spend on this exercise work.
3. **Select a** subject for your exercise work. Ensure that the chosen topic is not exactly the same as the ones already covered in previous exercises or presented by instructors in any course, including the current one. Aim for a well-organized and well-thought cohesive exercise work topic that encompasses the following key aspects:
   1. Multiple classes
   2. Multiple modules
   3. Multiple instances of a class
   4. At least one part of the work modeled using **UML sequence diagram**
   5. At least one part of the work modeled using **flow chart**
   6. At least one part of the work modeled using **UML class diagram**
   7. Some interaction/relationship between classes (association)
   8. Inheritance
   9. Objects passed as function arguments
   10. Integration of data structure (list, tuple, dictionary, etc.)
   11. Polymorphism

Also, ensure that your work is following the guidelines:

1. Adhere to the Style guide for coding consistency
2. Implement version control using Git
3. Provide well commented code
4. provide comprehensive documentation (see bullet 5)
5. Conduct testing (and evidence of testing visible in return document (test report))
6. **Keep in mind** your team’s goal of the exercise work:
   1. Easy(ish) (5-10 points): **The exercise work should demonstrate student’s ability to code classes in modules, create class instances and pass instances as function arguments.**  The work should be at least somewhat documented.
   2. Medium (11-30 points): Similar to easy(ish), and most of the requirements listed in bullet 3 should be evident in the exercise work. **The Exercise work should demonstrate that students can confidently apply the essential techniques of object-oriented programming.** The work should be well documented.
   3. Challenging (31-40 points): **All topics listed in bullet 3 must be covered.** The exercise work should demonstrate that the student has mastered all the topics covered in the OOP course and exercise topic is not entirely trivial. The work should be very well documented and tested, with adherence to all provided instructions.
7. **Return** your work biweekly in Itslearning. Modify the return template accordingly each week.
   1. **Part 25.3. – 6.4.2024:** at least preliminary topic, schedule of your project (what to do and when), goal (bullet 4), plan how to achieve the goal, possible missing tasks and topics listed (bullet 1) and how many % of exercise tasks are done, time spent on weekly exercises (bullet 2a), time spent on this exercise work part 1 on this week (bullet 2b). Topic is described and some coding (hopefully 25-50 %) has been done. Return the return document covering all the mentioned aspects (including the self-assessment) and take screen capture of the code and the outputs. Demonstrate to teacher.
   2. **Final – 7.4.2024:** final deadline: updated schedule (if changes), weekly hour tracking (bullet 2b), final versions of all diagrams, finished program, code well commented, all replacing task codes are clearly marked in code. Self-assessment how did you do (what went well, what challenges and successes you had, what would you still like to learn, are you happy with the outcome). Return the return document covering all the mentioned aspects (including the self-assessment) and take screen capture of code and running it. Prepare a 2-3 slides Power Point presentation (group members, topic of your work and a few screen captures of the outcome). Also prepare a short live demo (or video).
8. **Demonstrate your work in class either on 8.4.2024 or 15.4.2024**
   1. Prepare to demonstrate your work (keep the presentation). The demonstration is done in class for the whole class. You have 3-5 minutes / group for your demonstration. Make sure you have all your material and environments ready for the demo.