# Software Development

## Mini Assignment 3
Due: October 1, 2018 at 23:55 on myCourses

You are not writing a complete program for this assignment. Your solution program will only contain class definitions, data structure declarations, and method signatures. This partial solution program must follow well-designed principles and must reflect optimal decisions. You are not writing the body of the methods. You are not providing testing code. Your solution will be what is called a skeleton. The entire program is there but none of the methods have any code in them. A programmer could look at your skeleton solution and read the use-case, and from those two things, would be able to complete the program – or in our case evaluate if your design is good.

Look at this use-case:

Students in school Royal Blue come in two forms: full time and part time. Generally, these two types of students are identical except for the maximum number of courses they are permitted to take. Full time students must take between 4 to 6 courses per semester. Part time students must take between 1 to 3 courses per semester. These enrollment limits are strictly monitored. The following information is kept per student: name, student ID, history of courses taken (as references to course objects), and history of course grades (as an array of course reference and final grade pairs). The program has a main menu with the following options: (1) register students to a course, (2) add a grade to a student for a course, (3) create a student, (4) create a course, (5) quite program.

For the above use-case answer the following questions:

1. Provide a complete skeleton solution for the use-case. Given that you are not writing the code for the method bodies, your solution must still express the means to how you addressed all the important issues raised by the use-case through your selection of well-designed objects, good identifier and method names, data structures and method signatures, and all the OO techniques we covered in class for Java (to this point). Write this in Java.
2. Edit your solution, from 1 above, by adding Generics. If you like, you can merge questions 1 and 2, and simply return a single answer with Generics from the beginning. The question is split into two to help students focus on one issue (well-designed code) before addressing the other issue (Generics). However, if you can do both at the same time then do so.
3. Using your solution code from question 2, create a new “generalized” solution. A generalized solution is known as a Framework. The given solution code from question 2 can be generalized resulting in a solution that can be applied to multiple similar use-cases. For example: School could be replaced by Club, membership for registration, students could be different membership types, courses could be seminars or events or tasks, etc. A grade could be an award or points or a collection sizes, etc.. Modify your solution in question 2 by changing the names of all the classes, data structures and methods so that they do not refer to a specific use-case but can still be understood a providing the commands needed to solve a problem. Then, put this in a Package, so that this skeleton could be shared with others (assuming we provided code for the methods – which you are not doing). Make sure you maintain strong Generics to maintain the relationships correctly between your generalized structures.

To be clear, this assignment is asking you to write (1) a solution skeleton for a specific use-case, and then write (2) a sharable package that has generalized the given use-case so that other similar use-cases could use the package as a starting point for their application.

## WHAT TO HAND IN

- Three Java projects (or 2 projects if you combined question 1 and 2)
  - RoyalBlue project
  - RoyalBlueGenerics project
  - GeneralizedFramework package
- Zip everything into a single file

## HOW IT WILL BE GRADED

For your assignment to be graded your program (a) must follow the assignment instructions. You are doing this assignment on your own.

- +5 – Optimality (Big Oh and memory)
- +5 – Simplicity of solution algorithms
- +5 – Correctness of the application
- +5 – Uses encapsulation, information hiding, generalized code, code reuse, code similarity, APIs
- +5 – RoyalBlueGenerics project
- +5 – RoyalBlue project
- +5 – GeneralizedFramework package