# CS 1400 - Lab 5

Maximum Points: 10 pts.

## Lab Topics

- Classes and Objects in Java
- Class Methods and Attributes
- Understanding the use of permission modifier

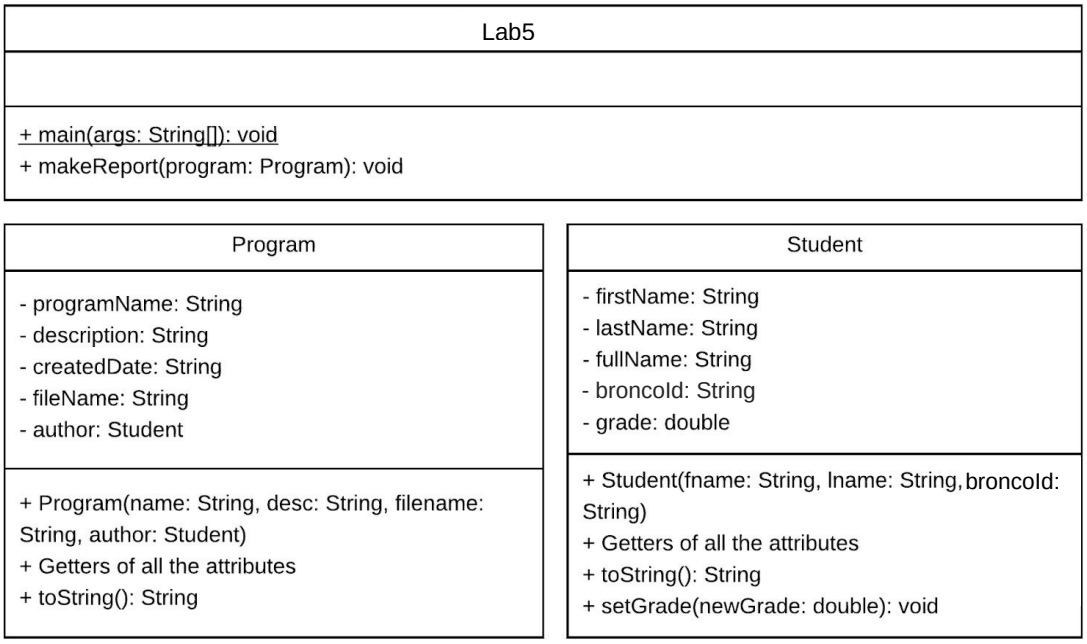
## Use the following Coding Guidelines:

1) Download the template file Lab5.java and Program.java from Canvas and fill-in-the-blanks to create your Java program. You will also need to create another file called Student.java.
2) Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
3) Keep identifiers to a reasonably short length.
4) Use uppercase for constants. Use upper camel case for classes. Use lower camel case for all other identifiers (variables, methods, objects).
5) Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches, and loops. Be consistent with the number of spaces or tabs that you use to indent.
6) Use white space to make your program more readable.
7) Use comments to explain how the parts of your program work.

## Lab Problem: Designing an AutoGrader

Grading computer programs is an extremely tedious and complex task in school. One core problem in this task is how to manage students' records and programs efficiently. In this and the following lab classes, you will design a real and working productivity tool that can help teachers to do grading in a Java programming class.

In Lab 5, your job is to define three classes that will become the building blocks of an auto grader application, which are Lab5, Program, and Student. Lab5.java is the main driver of this application. Program and Student are two custom data types (classes) that help the user to save and organize information in the application. The structures of these classes are shown in the following UML class diagrams (the details will be explained in lab classes).




Your program will do the following things:

1. Read a student's record and the program's details from the input
2. Create Student and Program objects to save the record into Java
    - Student grades are initialized with -1 (check the sample output)
3. Make a report in a structured format that can be filed in the future. Note that the report should be produced by a separate method, not by the main method.

## Hints & Notes

- The input function and some helper functions have been given in the Lab5.java file. You may still create your own Lab5.java as a practice.
- Read the main function in the given Lab5.java first. Try to understand what we expect the program to do and work.
- You will see a lot of errors in the original Lab5.java by default because variables and classes are not defined.
    - You will need to put in your own code in the main function in Lab5.java
    - You will need to create ONE class by yourself, Student (Student.java), and supply the required attributes and methods

# Sample Output

Below is an example of what your output should roughly look like when this lab is completed. Text in RED represents user input.

```
The student's first name? John
The student's last name? Williams
The student's Bronco ID? 102910295
Program name? CS1400 Lab 5
Program desc? Custom classes/data types for the AutoGrader project.
Program filename? Lab5.java
Program grade? 9

Making a student record
...[Student: John Williams, Bronco ID: 102910295, Grade: -1.00]
Making a program record
...[Program: CS1400 Lab 5, Desc: Custom classes/data types for the
AutoGrader project., Filename: Lab5.java, Author: Student: John
Williams, Bronco ID: 102910295, Grade: 9.00]

========== Program Submission Detail ==========
Student     : John Williams
Bronco ID : 102910295
Grade : 9.0

Program     : CS1400 Lab 5
Filename    : Lab5.java
Description : Custom classes/data types for the AutoGrader project.
Datetime    : 2020-10-08T15:37:36.188587
```

## Submit Your Lab5.java, Program.java, and Student.java to Gradescope

Please submit **ONLY** the java files to the “Lab 5” link on Gradescope. Make sure it is compiling and producing the expecting outputs. You are done.