[**Department of Computer Science**](http://www.cs.umd.edu/)

[**CMSC132:**](http://cs.umd.edu/class/fall2017/cmsc132/) Fall 2017

**Project:** Online Test System (Design)

**Due Date:** Wed Nov 1, 8:00pm

## Overview

For this project you will implement the data manager of an online test system. The system allows for the definition of exams with three possible kinds of questions: true and false, multiple choice and fill-in-the-blanks questions. The system will grade submitted exams and generate some statistical information.

## Objectives

* Practice the design and implementation of object-oriented software systems.

## Grading

* (45%) Design
* (40%) Public Tests Only (No release/secret tests)
* (15%) Command Line Interpreter

## Clarifications

## Code Distribution

The project's code distribution is available by [checking out the project](http://www.cs.umd.edu/eclipse/manage.html#check-out) named **OnlineTest**. The code distribution provides you with the following:

* A package named **cmdLineInterpreter** - Package where classe(s) that implement a simple command line interpreter for the System Manager will reside. We should be able to run your interpreter by running the main method of the **Interpreter** class. Additional details about this interpreter are provided below.
* A package named **onlineTest** - Package where all the classes representing your design will be implemented.
* A package named **tests** - Includes the public tests (**PublicTests.java**). Notice that you are not required to write student tests for this project; however, we recommend you do so.

## Specifications

### Data Manager

For this project you will implement the data manager (model component of the MVC paradigm) for an online test system. The system allows for the definition of exams with three possible kinds of questions: true and false, multiple choice and fill-in-the-blanks questions. The system will grade submitted exams and generate some statistical information.

The functionality of the system you will implement is represented by the interface **Manager**. The interface represents the methods available to anyone trying to make use of the system. The javadoc documentation for the project can be found at [javadoc](http://docs.google.com/doc/index.html). Your task for the design part of the project is to come up with a set of classes and interfaces that together provide the functionality defined by the Manager interface.

Access to the data manager functionality will be possible through a class named SystemManager that you must define. This class implements the Manager interface. Feel free to add any instance variables and private methods to this class. Public tests create an instance of the SystemManager class in order to test your system.

Your design should make use of object-oriented design concepts (e.g., generalization, specification, etc.) discussed in class. Keep in mind that having a large number of classes does not guarantee a better design grade. If you only implement the SystemManager class you will lose all the points associated with the design component of this project (even if the class provides all the expected functionality). We encourage you to discuss your design with your instructor or TA to make sure your design is a valid one.

## Requirements

* An exam can have different types of questions (true/false, fill in the blanks, etc.)
* We expect to see at least five classes/interfaces (SystemManager and four additional classes/interfaces) in your design. For this project, do not use anonymous inner classes (except for iterators). Notice that classes used for iterators do not count towards the minimum of five classes. In addition, classes associated with the command line interpreter are not included in this count.
* You should use maps/sets to keep track of your data.
* Your command line interpreter will provide access to a reduced set of the functionality of your system. You need to define a command line interpreter that allow us to:  
    
    
  + Add a student
  + Add an exam
  + Add a true/false question
  + Answer a true/false question
  + Get the exam score for a student

**Your command line interpreter will display a menu with the above options. Your interpreter should rely on standard input/output (not GUI/JavaFX).**

* Verify that your project passes the submit server tests (<https://submit.cs.umd.edu/>)
* Use the following approach to compute average: ((examScore1/totalScoreExam1) + (examScore2/totalScoreExam2) + ...)/totalNumberOfExams
* See [Student Tests](http://cs.umd.edu/class/fall2017/cmsc132/content/resources/StudentTests.html) for information regarding the implementation of student tests for this project.
* See [Style Guidelines](http://www.cs.umd.edu/~nelson/classes/resources/javastyleguide/) for information regarding style.

## Submission

Submit your implementation using the Eclipse "Submit Project" option (available by right clicking on the project folder).

## Academic Integrity

Please make sure you read the academic integrity section of the syllabus so you understand what is permissible in our programming projects. We want to remind you that we check your project against other students' projects and any case of academic dishonesty will be referred to the [Office of Student Conduct](http://www.jpo.umd.edu/).