MIT-AITI Design Exam II **July 13th, 2005**

This exam will test your practical Java programming and design skills. You are provided with a project specification and must implement a solution. Your exam solution will be graded on whether it meets the specified functionality, as well as its organization and clarity. Someone else should be able to understand, integrate, and maintain your solution by reading your code.

Partial credit will be given. It is recommended that you first write empty methods that return null or zero values before implementing the actual functionality. Your code should be properly commented and formatted. Put a brief comment at the top of each method explaining what you are trying to do. This will be especially helpful for partial credit if your program does not compile. It gives us an idea of what you are trying to accomplish. **But please try to make sure your program compiles at the very least.**

You may use any available resources during the exam, except your fellow classmates. You may not communicate with classmates in any way while the exam is in progress. **Add comments to your source code citing any outside material you use**. **Plagiarism will not be tolerated**. Students caught cheating will be immediately dismissed from the course and referred to the Dean.

Setup and Submission Instructions

- Create a new folder or project named with your first initial, middle initial and last name, for example "Alyssa Bourne Hacker" would name her project "abhacker". (This is similar to how you created your lab folders e.g Lab1)

- **In each class you implement**, include the following field with your name:
  `public static final String MYNAME = "Alyssa Bourne Hacker";`

- When you have completed the exam, drag the entire folder into the DropBox.

Contact a staff member if you have difficultly submitting your exam solution or need clarification on the exam description.

Problem Motivation

You are asked to create a small database for a Football team. You will then test it by modifying a file which contains information for 5 players, and writing this new information into another file. For this you will design a `FootballPlayer` class, and a `Gui` class used to add players to a roster and modify players' information from files.

FootballPlayer Class

Write a `FootballPlayer` Class. This class should have the players's name and yearly salary (in KShs). Players are instantiated with both of these. The `FootballPlayer` class should include a `getName` and a `getSalary` method which return the name and the salary of the player, respectively.

Gui Class

Create a `Gui` class to facilitate the addition and removal of players from the roster. The `Gui` class should also have the ability to read/write information from/to files. This should all be done in an interactive user interface. Your GUI should have an "**Add Player**" button, a "**Clear Roster**" button and a "**Write Info**" button.

When the "Add Player" button is clicked, the program should pop up a window to ask the user for the player name and after that is given, another pop-up to ask for the player salary. These shall be used to instantiate this player and add him/her to the roster. Exceptions should be handled if:

I) A player with the same name already exists in the roster.

The exception should cause a pop-up on the screen with the appropriate message.

When the "Clear Roster" button is clicked, the program should clear the roster (hint: check the API for a quick solution).

Run and test your code. Does it do what it is supposed to?

After testing your code, clear your roster. This is important because the next portion of the exam will need a specific set of 5 players.

I/O

You have been given a file called PlayerRoster.txt, available on \\chania\mit\exam2. This file contains information for 5 different football players. The information for each player is put on a new line. The following information is contained in each line of the file:

Player's Team, *Bonus Salary*, and *Number on the jersey*.

Note that only commas are used as the delimiters.

Using your GUI application, create the following 5 new players with the specified names and salaries below:

Name:	Yearly Salary(KShs):
Dennis Oliech	35,000,000
Musa Otieno	20,000,000
Francis Onyiso	30,000,000
Mike Okoth	25,000,000
John Muiruri	24,000,000

Writing information to a file

When the “Write Info” button is clicked the following should be done:

I) Add the yearly salaries from each of the 5 player objects above to the 5 respective Bonus Salaries in the file PlayerRoster.txt (i.e Dennis Oliech’s yearly salary should be added to the Bonus Salary in the first line of the text file giving him a Total Salary of KShs40,000,000, and so on).

II) In a new file (which you will create) called exam2.txt, write the following information for each player in a new line:

Player Name, Total Salary, Player’s Team, and *Number on the jersey.*

Does your exam2.txt contain the correct information in the right order?

Good Luck!
AITI-Kenya Team

EC.S01 Internet Technology in Local and Global Communities
Spring 2005-Summer 2005