



UNIVERSITY OF NICOSIA ΠΑΝΕΠΙΣΤΗΜΙΟ ΛΕΥΚΩΣΙΑΣ

COMP-455 Object Oriented Programming (Java) Fall 2009

Lecturer: Dr Constandinos Mavromoustakis
Email: mavromoustakis.c@unic.ac.cy

Project

INSTRUCTIONS:

Answer **ALL** questions. **Submit by email your program and clearly indicate the name of the course (COMP-455 OO Programming Language) and your name and student id number.**

The examiner will attach great importance to the quality of code. Marks will be deducted for poor programming style. The program should be commented and indented to aid clarity.

The program must not have any errors after the compilation phase, otherwise a grade zero will be assigned.

This is an individual project and should represent your own ideas, work and effort. In cases of plagiarism, a grade zero will be assigned to all parties involved.

SUBMISSION

In addition to your well-decomposed, well-commented solution to the assignment, you should submit a README containing instructions of how a user can run the program as well as specifying the number of available logged users that the system can handle. For the submission of the final version of the project a directory should be created with the NAME of each student and the ID number of each student in the form of NAME_ID (as Dir). When you are ready to submit, make sure that you have the above created Dir and the associated class(es) or files with your submission.

HINTS:

Your main program should work as a menu where all the required choices are displayed for the user to choose from. You should make sure that your program is as robust as possible. That is, it does not terminate with fatal errors unexpectedly in case of user error. It should validate input where necessary and allow the user to re-enter the required information.

Note: IF and only IF (IFF) you use additional classes from other authors, all should be clearly cited and clearly marked as borrowed from “X” author. Additionally these classes (or parts of code) will decelerate you from reaching the highest possible marks for each section and probably for the entire project.

Deadline is before the examination Saturday, Jan 16, 2010 at 10 a.m., by email/along with documentation.

No projects will be accepted after this deadline.

Project part(s) and Marking		
I	The GUI components/ GUI design/Operational characteristics/OO-based format	80%
II (optional)	Applet-based scenario (optional) /(Bonus 15% for the total grade of the Final Project)	15% (Bonus)
III	UML diagram explaining the entire program with model-based explanations	20%

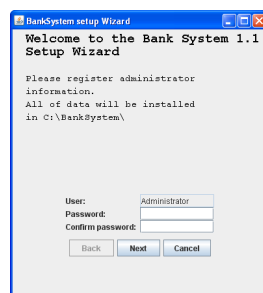
(I) (Note: Keep in mind that you should create your own scenario for any banking process)

This Project basically will focus on The GUI components and on the way that the following will be manipulated:

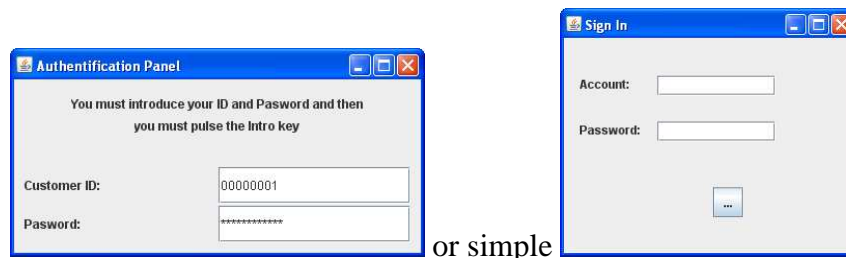
- InGUIs' environments to effectively handle events generated by user interactions with GUIs.
- To understand the packages containing GUI components, event-handling classes and interfaces.
- To create and manipulate buttons, labels, lists, text fields and panels.
- To handle mouse *events* and keyboard *events*.
- To use layout managers to arrange GUI components
- The usage of ALL characteristics that Java programming Language gives us.

We need to develop an application that utilizes/emulates and simulates the behavior of a number of small tasks which occur in a banking system. Mainly we need to develop a GUI (Graphical User Interfaces) basically by using the design principles of GUIs. The application must have a user friendly approach and should have conversational characteristics (dialog Boxes) for the end-user's interaction with the system. The system will consist of many clients and one server machine. The server machine is entirely responsible for the authentication of each one of the end users. The program should also have a Driver (master class) to exercise and drive the use of multiple Account objects.

The system should ask the user to enter an ID.



OR



or simple

Then the system authenticates this ID. Thereafter the MENU is presented in the following form:

Customer ID:

NAME:


Address:

Account type: Ordinary/Parental/ Dormant

Available Balance:


Cards:

Loan(s):

Arrears: (in this case if true a warning button  should appear)

On a *tabbed format (which will be developed)* on the same card, the user should be able to change information like address or correcting name. For changing these attributes the system must ask the end user for a second password (which should be written on the screen in a form of *****). Tabbed format should be used for 3 (three) other attributes: (i) Transaction (which will include all the history of the Transactions made for the specified account/person), (ii) sub-accounts -which includes on the same format as above- the sub-accounts for this particular customer and (iii) Funds Transfers which includes Transfers To Other Banks & CO-Ops, Settlement of Account, Payment Orders, New Payment Order, Pending/Completed Pos and Companies' Payroll. Operations for the (ii) Sub-accounts and (iii) Funds Transfers are free of guidance. It is strongly recommended that you should use your imagination for some extra functions, and will be considered as an *extra bonus* 10%.

In case of Account type: ~~Ordinary~~/Parental/~~Dormant~~ ->The Parental type should enable the pop-up of an extra dialog window with the same information as the initial, and also should include the account or accounts of his/her children (if any).

If there is something that the end users should be informed about, a GUI win should inform him/her with an  icon in the dialog Box. All dialog Boxes should be supported by a user-friendly interface.

Also there must be a print out facility and also a file backup facility for the end user. For the print out facility, you can only set a button, but the printouts should be in real time (the required document(s) or data should be printed). The backup facility should backup the entire account

(account details) in a file called account.bak. Additional grade-bonus of 5% will be given, if a name-it option is supported, in which the end user is asked for the name of the backup file.

Note: You should make use of all the OO concepts we have been over (Objects, GUI, Inheritance, Overloading, Polymorphism etc.)

You need to submit :

- *The source code and the executable along with a precise documentation explaining the given code and how your code operates.*
- *A clear definition for the source code regarding the Constructor methods and the actions with the associated members/functions.*
- *All source codelines should have comments where needed.*

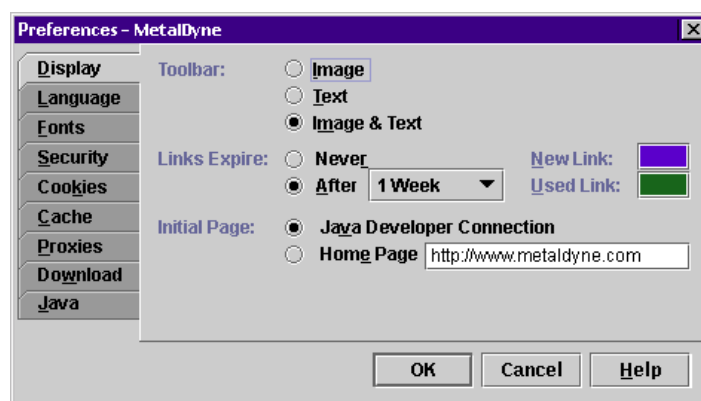
(II) Applet-based scenario (optional):

(Bonus 15% for the total grade of the Final Project)

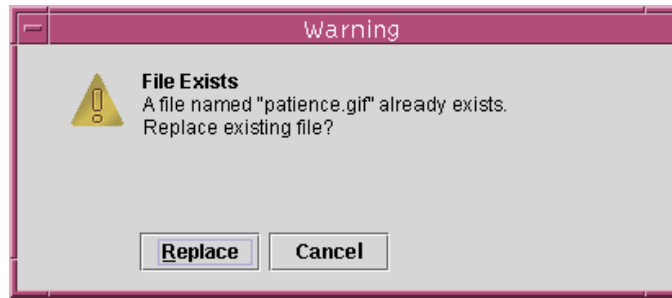
Develop the above application on a web based interface and format. Classes should not significantly change in order to offer the same operational characteristics.

(III) Finally for each one of the above programmed operations (I) (and optionally (II)), you should provide a UML diagram explaining the entire program with model-based explanations (where required) for the reverse engineering process.

Remarks: You should design/write, create and use a GUI (Graphical User Interface) template. GUI design is also an important part of application programming. As an idea of what is expected for the user friendly interface, are the fig. windows below:



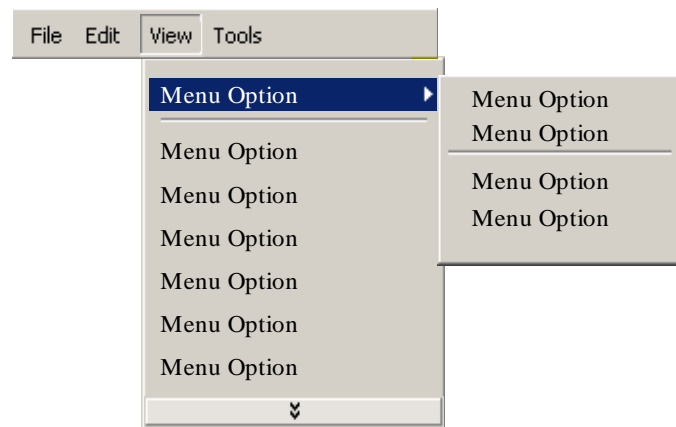
An example of a possible warning message:



An example of possible set of buttons:



Main win MENU type:



Notes:

- All above conversations will have an 1-1 interaction with the main server in order to keep the authentication criteria satisfied. All database attributes can be kept in data structures in server's side.
- You are strongly enforced to refer to Sun's Java official web page for the GUI components as well as the custom use of their constructors. (<http://java.sun.com>)