Class Models for CSC 505

See examples below for information on how to write your class and class operation forms. You need to write a form for every class in your system. (If you are not using an object-oriented language, then a form needs to be done for every major module or script or function.) If you have not already, you can begin to write the code for some of the most important functions and begin to test them. Use comments to supply any "non-code" information. Place each class on a different page please. It is okay to hand this in missing some or all of the function implementations. Notice that some relationships that were in your class diagrams will now be represented as classes and some distinct classes will be collapsed into the same class, etc. These models are now very close to the actual implementation classes and modules and functions (contrary to the class forms that reflected a conceptual model and not necessarily the implementation model).

Important: On each class form, please attach a symbol that indicates which team member prepared the form.

Don't hesitate to ask if you have further questions.

Grading

Completeness of Class Forms:

___ 20% Complete form for every class (major module, script, and function)

___ 20% Reasonable level of detail

Consistency among all forms and diagrams and readability:

___ 30% Documentation of class forms and comments

___ 15% Correlation with CRC cards and (OR) object-relationship diagrams

___ 15% Correlation with scenario diagrams and the methods/relationships of the CRC cards and the relationships of the OR diagrams.

Sample Class Description Form

Name	*Mailbox*
Base Class(es)	
Purpose	*A mailbox contains messages that can be listed, kept, or discarded.*
States	*Empty, full, or neither.* *Inactive or Active*
Constructors	*Default: makes inactive mailbox*
Operators **Mutators** **Accessors**	*Receive_message(Message)* *login():process password and commands* *retrieve_messages()* *delete_current0* *change_greeting()* *change_password()* *activate(int extension, int passcode)* *Message get_current()*
Fields	*MessageQueue_new_messages* *MessageQueue_kept_messages* *String_greeting* *_extension* *_passcode*

Sample Method Description Form

Prototype	*Bool MailSystem::add_mailbox (int extention, int passcode)*
Purpose	*Add a new mailbox to the system*
Receives	*extension-the extension number of the mailbox* *passcode-the passcode number*
Returns	*TRUE if the operation succeeded*
Remarks	*The operation may fail if the extension number is already in use or if there I sno space to add another mailbox.*