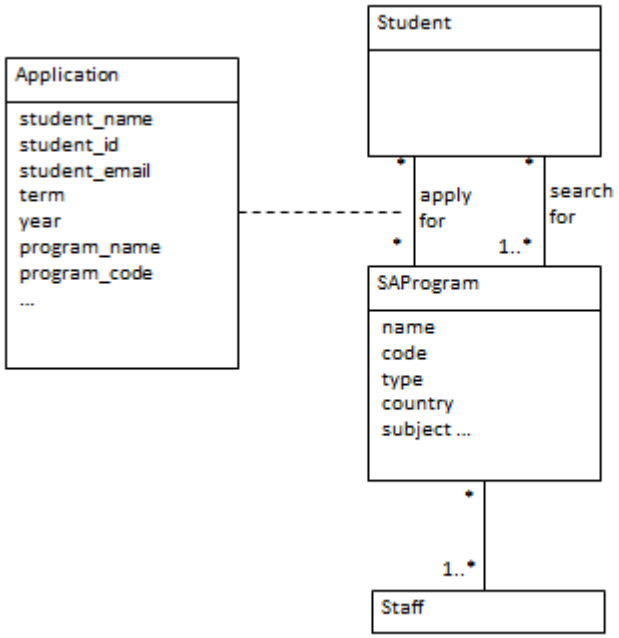
2014F CMPS115
Quiz 3 **Name**:_____Points_____/100

System being developed: Study Abroad Management System (SAMS)
SAMS manages a collection of study abroad programs that students can search and apply for, and staff updates.

Here is part of its domain model:



User Story 7: Add Program
As a staff member, I need to add new programs to the Study Abroad (SA) catalog, so students can have a wider choice of programs.

Here is an expansion of User Story 7:

Expanded User Story 7: Add program

precondition: User is logged in as a Staff member	
Actor: Staff User	**System**: SAMS
	0: System displays welcome page
1: **TUCBW** staff user clicks on Program Management link on welcome page	2: System shows the Program Management submenu
3: Staff user clicks on Add Program link.	4: System shows the Add Program Form with name, code, type, country, …
5: Staff user fills the form and clicks on Submit button	6: System displays a "program successfully added" message.
7: T**UCEW** staff user sees the "program successfully added" message.	
postcondition: added program is immediately available in searches by students	

Question 1: [20] Expanded User Stories/Use Cases

a) [10] Explain the format (the components) of an expanded User story/Use case.

An expanded user story expands a user story into a dialog between a user (in certain role), i.e. the actor, and the system (under development). It therefore it has these components:

1) A name that identifies the user story it expands.
2) If needed, a precondition specifying relevant aspects of the system state that must hold before the user story begins.
3) A table, consisting of two columns: one for the actor, one for the system.
4) A step 0, describing the state of the user interface the actor observes before the dialog begins.
5) A marker for the step at which the user story begins (TUCBW), usually step 1, an action by the actor.
6) A table row for each request/response turn between actor and system.
7) A marker for the step at which the user story ends (TUCEW), usually an action or observation by the actor.
8) If needed, a postcondition specifying properties of the system state that must hold at the end of the user story.

b) [5] What is the purpose of an expanded user story?

Expanded user stories refine the high-level user stories into exchanges of specific user actions and system responses that the user can initiate and observe, respectively, via the system's user interface. This more concrete form of user story helps clarify and validate the requirements; it provides guidance for the system design and increases the confidence of both users and developers that the right system is being built.

c) [5] Expanded user stories are useful for creating what kind of test and what kind of documentation? Briefly explain. [Hint: consider what information they provide; do they take an inside or outside view of the system?]

1) User's Manual.
   i. Expanded user stories can serve as (the basis for) a preliminary user's manual, which can aid users in experimenting with early versions of the system.
   ii. If kept up to date during development, expanded user stories are a basis for generating an as-built user's manual, reducing the effort at deployment time.
2) Black box and acceptance testing.
   Expanded user stories include only phenomena observable at the system's user interface. They therefore can serve as a basis for black box and system acceptance testing.

User Story 8: Apply for Program
As a student, I need to be able to apply for a study abroad program, so that I can take full advantage of my educational options.

Question 2: [25] Expand User Story 8: Apply for program

a) [15] Expand User Story 8, under the following assumptions:

i. To apply to an SA program, a student must be logged into SAMS.
ii. Each page of SAMS displays a navigation link "Apply for program".
iii. When the "Apply for program" link is clicked, SAMS displays a form with fields described in the Application class of the domain model.
iv. After application form submission and form data validation and evaluation, the system responds with a message reporting success or failure of the submission.

In case of a successful submission, the student will receive a confirming email message.

Expanded User Story 8: Apply for program

precondition: User is logged in as a student.	
Actor: Student	**System**: SAMS
	0: System displays some page with navigation menu.
1: **TUCBW** student clicking on navigation link "Apply for program".	2: System displays application form with fields including student name, id, and email, and program name and code, etc. **Trivial**
3: Student fills out form and clicks submit button.	4: System displays message informing whether submission succeeded. **Non-trivial**
5: T**UCEW** student seeing message reporting success of submission.	
postcondition: if submission was successful, an email is sent immediately to the student confirming submission.	

b) [5] For non-trivial steps in an expanded use case it is useful to develop diagrams detailing the system-internal object interaction necessary for the system response.
Mark each actor/system exchange in your expanded user story as trivial or non-trivial. Briefly explain why.

Step 2 is trivial because it is the same for all students and requires no background processing.
Step 4 is non-trivial because it requires data base access for both data validation and processing, e.g. recording the application.

c) [5] The domain model captures the concepts used in (expanded) user stories. Give three attributes the class Student should provide. Briefly explain why.

Three attributes the class Student should provide are name, id, and email. These items are part of the student related data in the application that need to be validated upon submission against the student records.

Question 3: [30] Object Interaction
The object interaction diagram below specifies in a somewhat informal way the processing of an application in SAMS upon submission by a student.

a) [15] Number the arrows in the diagram in the order messages are sent and responses are received.

b) [10] Briefly describe the 5 exchanges initiated by the :AppController object.

1) :ApplController calls :Student to validate the form data
2) :ApplController calls :DBMgr to create SAProgram object for the program with code sap_code.
3) :ApplController sends the application data to the newly created program object for evaluation.
4) If the application is accepted, :ApplController creates a new application object, and
5) sends it to be saved by :DBMgr.

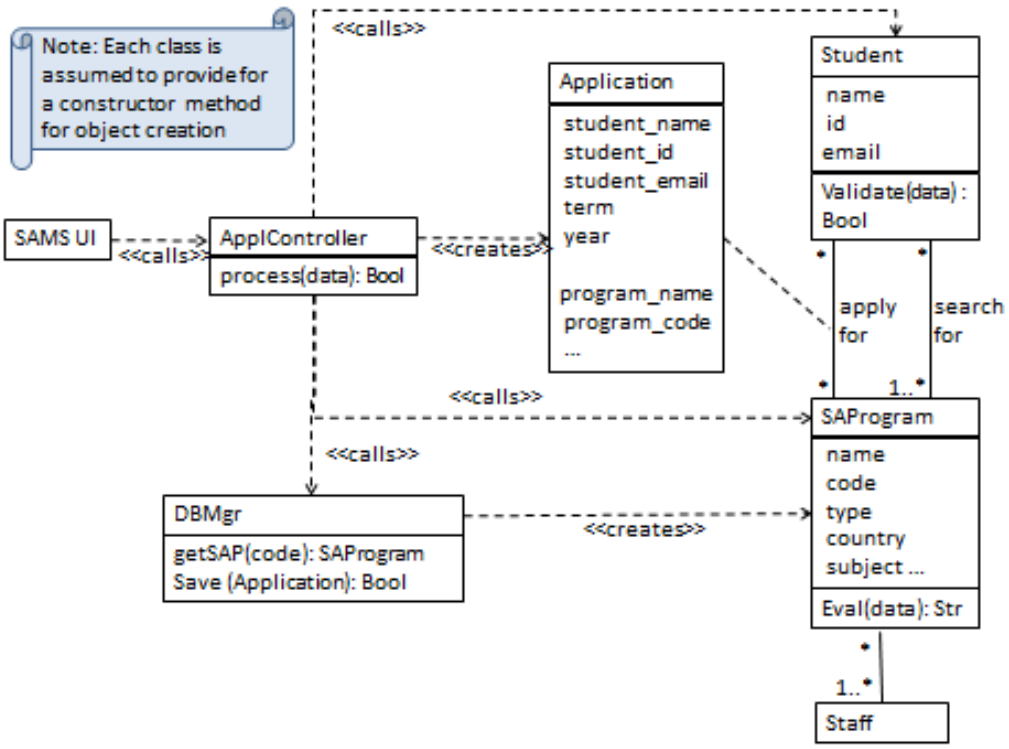
c) [5] In the diagram, mark the primary object or arrow involved in a controller pattern (CO), information expert pattern (IE), and object creation pattern (OC).

Question 4: [25] Create a design-level class diagram for SAMS.

a) [10] Extend the domain model to include all classes referenced in the informal object interaction diagram.

b) [10] Introduce methods and allocate them to the appropriate classes to support the message response pairs between objects in the informal object interaction diagram (each of these pairs will turn into a method call). Be as specific about the parameters and result types as possible.

c) [5] If, based on the informal interaction diagram, an object of class A calls a method of class B, then show this relationship with an arrow from A to B marked with <<calls>> in the class diagram.
If an object of class A causes the creation of an object of class B, then show this relationship with an arrow from A to B marked with <<creates>> in the class diagram.



2014F CMPS115
Quiz 3 **Name**:_____Points_____/100

Notes to diagram:
smiley: same as stick figure
appl_data: the data submitted in the application form; something like a Python dictionary
sap_code: the SA Program code specified in the application form (see domain model).
DBMgr: data base manager
Appl: same as Application
ApplController: the controller for application processing
SAMS_UI: graphical or web-based user interface to SAMS;
[condition]: means the message is sent only if the condition is true;
this diagram intentionally omits all error handling.