# University of Toronto

**Faculty of Arts and Science**
**Dept of Computer Science**

## CSC302S – Engineering Large Software Systems

**April 2008**
**Instructor:** Steve Easterbrook

---

**No Aids Allowed**
**Duration: 2 hours**
**Answer all questions.**

**Make sure your examination booklet has 10 pages (including this one). Write your answers in the space provided.**

**This examination counts for 35% of your final grade.**

---

Name: _____
(Please underline last name)

Student Number:_____

**Question Marks**

1_____ /20

2_____ /20

3_____ /20

4_____ /20

5_____ /20

Total _____ /100

## 1.      [Short Questions; 20 marks total]

**(a) [Software Quality – 5 marks]** What does "quality" mean with respect to software, and how would you measure it? Why is it difficult to predict the quality of software before you have finished building it?

**(b) [Verification Strategies – 5 marks]** What is the difference between testing and static analysis? What are the advantages and disadvantages of each of these approaches to verification?

**(c) [Requirements Analysis – 5 marks]** Why is it useful to write down the requirements for a software project before attempting to design a solution? Why is it hard to do this well?

**(d) [Agile Development Practices – 5 marks]** Agile practices are often considered to be appropriate only for small teams (e.g. less than 10 people). Which aspects of agile development can still be used effectively with larger teams, and which aspects fail to scale up?

2.      **[Black Box Testing – 20 marks]** You have been asked to test a program that takes as input three integers that represent the lengths of the sides of a triangle. The program outputs a string giving the type of triangle: "Equilateral" (if all three sides are equal), "Isosceles" (if 2 sides are equal), "Scalene" (if no sides are equal), or "Error" (if the inputs cannot be a triangle). What test cases would you use to test this program as thoroughly as possible? What else would you need to know in order to improve the thoroughness of your test suite?

3.      **[Class Diagrams – 20 marks]** The University of Kinakuta (UoK) has decided to implement an electronic card system for its students, so that students can use their K-cards to access secure areas (.e.g labs and residences after hours), and also as a debit card, linked to an account into which students can deposit money, to be used to pay University fees at locations on campus. For the initial release of the system, this will be limited to a few university fees: parking fees at campus parking lots, library fees at campus libraries, and equipment rental at the sports centre.  The system will keep a usage record for each K-card. An initial domain model for the system has been produced:
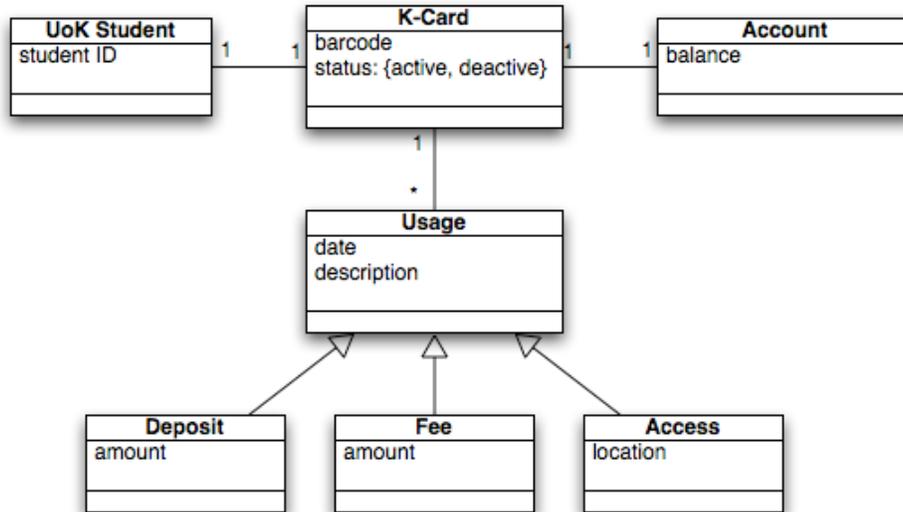


**(a) [5 marks]** In the model provided above, there is a 1-to-1 association between a student and a K-Card. Is this an appropriate way to model students and K-Cards or is there a better way? Defend your answer.

**(b) [5 marks]** In the model above, there are no explicit associations from the Deposit and Fee classes to the Account class into which these transactions are to be made. Should such associations be added? Defend your answer.

**(c) [5 marks]** In the original model, the relationship between the Usage class and K-Card class is an association. Alternatively, this relationship could be modeled using an aggregation or composition. Which alternative is better for this problem domain and why?

**(d) [5 marks]** After further discussions, UoK decides to add more features to the system. Food services wants to keep track of each student's meal-plan allowance when they use their K-cards to buy food and snacks at all campus eateries. Computing services wants to keep track of each student's printing allowance, when they use their K-cards at self-service printers across campus. The meal-plan allowance and printing allowance are separate from the debit-card balance. Student will also be able to use their card to make purchases (as well as paying fees) and to make deposits at many locations across campus, and the system will need to keep track of the location of each transaction. Revise the original domain model to allow for these additional features. Cross out elements that are no longer needed, and add any new elements. Use multiplicities, associations, generalizations, etc as appropriate. You may use the space below to re-draw the entire model if necessary.

---

4.      **[Topic – 20 marks]** Describe *two* examples of software engineering techniques (or tools) that you used on your course project this term: *one* that was particularly effective, and *one* that was not. In each case, explain how you used the technique (or tool), and the factors that caused it to be useful (or not). Your answer should take into account a consideration of whether the technique/tool was suitable for the type of software you were building, whether you used it at the right point in your development process, and whether you had the right expertise to use it properly.

5.          **[Software Communication – 20 marks]**. Imagine that a newly hired programmer is to join your team to assist with the work you are doing of adding new features to the UMLet application. Draw some pictures to explain to your new teammate the structure and behaviour of UMLet, and the feature(s) you have added in your course project. Your pictures should concentrate on key aspects of the design that would be most useful for the new team member. Assume your new team member is familiar with all of UML, but knows nothing about the UMLet application.

[scratch paper]

[scratch paper]