CS 247 Spring 2014
Assignment 1 Deliverables

Assignment 1 deliverables are worth 5% of your overall course mark.
No groupwork is permitted on this assignment.

Due dates

The project itself and the UML model are due on **Friday May 21, 2014** at **12:00 noon.** They are to be submitted electronically through **Marmoset**.

Marking Rubric

Programming questions will be marked for design quality and programming style as well as correctness. See the marking rubric for details on how your programs' design and programming style will be marked.

Q1 [40 marks] Polymorphism

You are to complete the implementation of a family of ADTs for cellphone accounts.

Submission

Add your implementation of the `Account` ADTs to the provided file `AccountTestHarness.cpp`. ***Do not change any of the provided code, do not change the public interface for Account, and do not include additional libraries.*** Submit the changed file electronically to Marmoset.

We provide an executable `Accounts`, which you can use to compare the output of your program against. The executable runs on the Linux machines in the undergraduate environment.

Marking

The test scripts and marking scheme are structured according to the following increments.

1. Commands **E, C**
2. Command **P**
3. Command **b**
4. Command **p**
5. Commands **c, B**

We strongly recommend that you implement each increment to completion (for multiple `CheapAccounts` and `ExpensiveAccounts`) before progressing to the next increment. ***The output of your program with be checked automatically using Marmoset — it must match exactly the output of our solution.*** You can check this by using the UNIX command `diff` to compare the output of your program against the output of the provided executable.

Q2 [55 marks] Copy, Assignment, Equality

You are to complete the implementation of an ADT for dynamically-sized lists of strings. The size of a list grows whenever an element beyond the last element in the list is accessed (that is, `operator[]` resizes the list).

Submission

Add your implementation of the `DynList` ADT to the provided file `DynListTestHarness.cpp`. ***Do not change any of the provided code, do not change the public interface for data representations of class DynList , and do not include additional libraries.*** You may change the public interface for Node or make the DynList class a friend of Node. Submit the changed file electronically to Marmoset.

Marking

The test scripts and marking scheme are structured according to the following increments.

1. Command **x**
2. Command **e**
3. Command **c**
4. Command **d**
5. Command **a**

We strongly recommend that you implement each increment to completion before progressing to the next increment. ***The output of your program with be checked automatically using Marmoset — it must match exactly the output of our solution.*** We provide an executable `Lists`, whose output can be compared against the output of your program.

Q3 [20 marks] ADT Design

You are to design an ADT for valid values for UW course marks.

Submission

Put your answers in a text file titled `ADT.txt`, and submit the file electronically to Marmoset.