CS 247 Spring 2014
Assignment 3 Deliverables

Assignment 3 deliverables are worth 5% of your overall course mark.
No groupwork is permitted on this assignment.

Due dates

The questions are due on **Monday July 18, 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 [70 marks] Design Patterns

You are to create a new example program that implements one of the specified Design Patterns.

Submission

Implement your example application in C++. Provide a Makefile that builds the executable of your example program, where the name of the executable is `exec`. The TAs will execute your program.

Place your answers to questions Q1a and Q1b in a PDF file named `Assign3.pdf` and ***submit it to the Marmoset Q1ab dropbox that corresponds to the pattern that you implemented***.

Zip the files (not the directory) that make up your answer to question Q1c as a zip file called `Pattern.zip` and submit it to the Q1c Marmoset dropbox.

Q2 [20 marks] STL

You are to provide code fragments that use STL algorithms to implement snippets of functionality from the Straights project.

Submission

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

Q3 [60 marks] Generic Programming

You will use C++ STL containers and algorithms to implement a Hangman program.

Submission

Place your program in a file called `hangman.cpp` and submit it to Marmoset.

Marking

The test scripts and marking scheme for this question are structured according to the following increments:

1. read input file of suitable words and output words to file named `gamewords`
2. read input file, remove unsuitable words, and output suitable words to `gamewords`
3. prepare pool of lowercase gamewords, and play hangman
4. prepare pool of mixed-case gamewords, and play hangman

We strongly recommend that you implement each increment to completion before progressing to the next increment.

The output of your program will be checked automatically using Marmoset — it must match exactly the output of our provided executable. We provide an executable `hangman` that runs on the Linux machines in the undergraduate environment. You can check your code against the provided executable using the UNIX command `diff` to compare the outputs of the two programs.