CS 247 Spring 2012
Project 1 Deliverables

The project 1 deliverables are worth 7% of your overall course mark.

1. Due dates

The project itself and the UML model are due on **Monday June 18, 2012** at **5:00pm.** They are to be submitted electronically through **Marmoset**.

If you choose to do this project as part of a team of two, you need to register your team by **Thursday June 7,** at **10:00am**. Team registration is done using the **submit** command.

2. Teams

This project may be done individually or in a team of two. If you choose to work in a team, you must register your team and its membership by **Thursday June 7 at 10:00am.** To register:

1. Create a text file called `group.txt` containing the user IDs of both members:

   ```
   userid1
   userid2
   ```

   Make sure that the file is only two lines in length, and that it uses the Unix newline format (as opposed to Windows)

2. Log into a Linux machine, then submit using the following command (where the "." means that the group file can be found in the current directory)

   ```
   submit cs247 group .
   ```

Only one person in the group needs to do this. After the group submission deadline, new Marmoset accounts will be created based on the groupings. Note that you will be in the same group for part 2 of the project.

3. UML Model

Because this course is focused on design of ADTs and medium-sized programs, you are to submit a UML class diagram of your program. It must include

- A class for each of your ADTs (include `main()` as a "class")
- A complete specification of all the attributes and operations in your human, computer, and generic player ADTs
  - e.g., name, type, initial value, static (if applicable), visibilities, abstract
- All associations between ADTs, and between `main()` and your ADTs
- All multiplicities that are not 1
- All navigabilities
- Composition and aggregation "diamonds", where appropriate
- All inheritance relationships

Name your UML diagram **uml.pdf** and submit it through Marmoset. We recommend that you start with the UML Diagram, just so that you think through the program before diving in.

4. Source Code + Makefile

Submit all your source code through Marmoset in a zip archive named **straights.zip**. It should include

- All .h files (including any provided .h files)
- All .cpp files (including any provided .cpp files)
- Makefile

Your makefile must create an executable called `straights`. Marmoset will run this executable.

5. Marking

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

1. quit command
2. shuffle (with and without random seed), deck command
3. human players only
   a) print player's hand (identify first player, print legal moves)
   b) play command (legal and illegal plays)
   c) discard command (legal and illegal discards)
   d) scoring
4. computer players
   a) play command
   b) discard command
5. ragequit command

We strongly recommend that you implement each increment to completion before progressing to the next increment. This ensures that you always have a working program to submit, and it will help you to maximize the number of test cases that your program passes should you not complete the entire project.

We provide an executable `straights`, whose output can be compared against the output of your program. ***The output of your program with be checked automatically using Marmoset — it must match exactly the output of our provided executable.*** You can check this by using the UNIX command `diff` to compare the output of your program against the output of the provided executable. The executable runs on the Linux machines in the undergraduate environment.

In addition, just like your assignments, we will grade your program based on its design and programming style as well as its correctness. See the rubric for details on how your program's design and programming style will be marked.