CS 580 Client-Server Programming
Spring Semester, 2010
Assignment 1

San Diego State University -- This page last updated 1/20/10

Assignment 1

Due Jan 29 23:55

1. Create a EvenQueue class that has the methods enqueue and dequeue. The enqueue method throws an exception when its one argument is not an even integer. When its one argument is an even integer, the integer is placed on the queue. When dequeue is called on an empty queue it returns null. Add your code your Mercurial source code repository. Your EvenQueue class can use collections classes from your languages class library.

2. Modify your queue in 1 to throw an exception when dequeue is called on an empty queue. Add this modified code to your source code repository.

3. Modify your queue in 1 to ignore arguments of enqueue that are not even integers rather than throw an exception. Check your code in as a second branch to your source code repository from your commit in problem 1. This branch does not contain the changes made in part 2.

4. Merge the two branches in your repository to create a queue that meets the requirements of 2 & 3. Merge version 2 into 3 creating a fourth version.

The goal of this assignment is to practice using a source code repository. Given a source code repository you need to be able to:

- Check code in
- Check out different version of your code
- Create branches of your code
- Merge different branches of code and resolve any conflicts.

Mercurial

Mercurial is a distributed source control system. The main Mercurial web site is http://mercurial.selenic.com/. It runs on Mac OS X, Windows, Linux/Unix. The Mercurial web site contains a download page. Instructions to install the Mercurial Eclipse plugin are at http://www.vectrace.com/mercurialeclipse/. If you use Eclipse you should also install the Mercurial runtime for your OS. I have found that some operations are easier using the Mercurial runtime. All the mercurial documentation uses the command line version of Mercurial. Also I have found that pushes work better in the Mercurial runtime than using the Eclipse plugin.

Mercurial Documentation

Short Tutorial: http://mercurial.selenic.com/guide/

Longer Tutorial: http://mercurial.selenic.com/wiki/Tutorial

Mercurial Book: http://hgbook.red-bean.com/

Things to do before Tuesday January 26

You should install Mercurial on your machine, both the OS version and the Eclipse plugin if you use Eclipse. Create your BitBucket account (see How to turn in your Assignment below). Read the documentation to understand the commands:

init, add, status, commit, log, update, merge, resolve, clone, push and pull

Using your favorite IDE or editor you should create a file, put it under Mercurial source control, commit several changes to the file. Push the project to your BitBucket account and clone a local copy from your BitBucket account. Load the cloned local copy into your IDE or editor to see that you can run the code and see the different versions of the code. You need to get comfortable making changes to your code, pushing the changes to a central repository and then deleting your local copy.

How to turn in your Assignment

You need a Bitbucket account (http://bitbucket.org/) for your course assignments. Sign up for the free account. Create a private repository for your assignment. Give the user rogerwhitney read/write access to your repository. Add your assignment to the private repository. Make sure that all parts of the assignment are in the repository. When your assignment is done you need to turn in the url to the repository for the assignment at the course portal. Your assignment is not turned in until you have provided this url. The time you provide this url determines if your assignment is late. When you provide this url the course portal downloads the project. If your project has a url of:

```
https://uerName@bitbucket.org/userName/projectName/
```

the course portal downloads the code using using the command:

```
hg clone https://uerName@bitbucket.org/userName/projectName/
```

Each time you submit a url for the same assignment the course portal removes the old files it cloned and clones a fresh copy of your project. Note this means if you submit the url before you push your assignment to BitBucket the course portal will not get the correct version of your code. You are strongly urged to use the above command to clone your BitBucket project to make sure it is correct.

Grading

Code Compiles & Runs	20%
Code downloads correctly from BitBucket	20%
Code has required versions	60%