COMP 322: Fundamentals of Parallel Programming (Spring 2014)
Instructor: Vivek Sarkar
Worksheet 14: due by start of next class

Name: _____ **Netid:** _____

Honor Code Policy: You are free to discuss all aspects of in-class worksheets with your other classmates, the teaching assistants and the professor during the class.

HJ-Lib Module 1 APIs

Consider the following HJ-Lib APIs:

1. `async(HjRunnable runnable)`
2. `asyncAwait(HjFuture<? extends Object> f1, HjRunnable runnable)`
3. `asyncSeq(boolean sequentialize, HjRunnable runnable)`
4. `doWork(long n)`
5. `finish(FinishAccumulator f1, HjRunnable runnable)`
6. `forall(Iterable<T> iterable, HjProcedure<T> body)`
7. `forasyncChunked(int startInc, int endInc, HjProcedure<Integer> body)`
8. `future(HjCallable<V> callable)`
9. `futureAwait(HjFuture<? extends Object> f1, HjCallable<V> callable)`
10. `next()`

For each of the following functionalities, enter the number of the API above that matches the functionality:

Functionality	API number
Delays execution of an asynchronous task until a specific value becomes available, but does not return a handle	
Spawns an asynchronous task and returns a handle which can be queried to determine if the spawned task has completed execution	
Does not include an outer finish while parallelizing a loop	
Can be used to tune the parallel program for performance	
Spawns an asynchronous task which may run in parallel with the parent task	
Creates a parallel version of a for loop	
Is commonly used in HJ programs which need barriers	
Delays execution of an asynchronous task until a specific value becomes available, and also returns a handle	
Used by abstract performance metrics to report work done by the currently executing task	
Enables accumulation to be performed safely in a async-finish style program	