TestingThreads is class originally worked on. It just uses ArrayBlockingQueue and Executor (thread pooling) to basically mimic producer/consumer. Except producer is request getting sent to it, doesn't generate its own.

---issue is in this setup, thread can't return values and don't see way to send info back to correct user since we put them in queue then took them off. You would call submitter to put message in queue and then that’s it, no method call traceback [I think]. So, unless front end has a special method back end can use to directly send info back, this version is probably lost cause (specifically queue part)

TestingThreadsWithCallBack

Is similar to TestingThreads but doesn’t use BlockingQueu. Instead uses queue built into executor which is infinite. So, could send request by calling method requestConsumer and thread can return info due to callback and future. [Hopefully this works, not sure if static would affect it. Only reason have it is because a tutorial video did it.]

--Note callable doesn’t have to be type string, can be any. Just need to stay consistent. Future type needs to be same as callable type

AnotherClass and SomeClass is supposed to mimic classes that would deal with request like Log in, Posting Swoop, Getting swoops, etc. NOTE: Have to use framework that these classes have. The Callable and call method. Call is method that will be called when request obtains a thread from pool. Callable allows thread to return values while Runnable does not.

RequestObject is an example of what we may have to format request since we need a number identifier so we properly thread it

BackEnd class is class that has method you call to send request to backend, it also setups up threads for each request and will call appropriate class for that request using the number identifier

Can delete all system.out.println, just left them just to show how it works

In TestingThreads, made 4 threads to simulate users trying to send request to back end practically at same time. Due to how threading works, the print statements order are not fully accurate. From testing it seems code works fine. Multiple users could send request to back end and retrieve proper data back. Speed of this depends on how long request takes to execute (other class codes). Could reduce waiting time somewhat by increase thread pool size.

Note: mainly should be concerned with BackEnd class, the other classes are just to show how should set up everything for backend code to work properly. Of course code will need edit to fit other classes, such as switch statement in BackEnd

99% certain the request are being sent to same instance of class BackEnd despite note creating a set BackEnd object. Previous test showed queue was changing in size when had setup of multiple threads doing request. Would be 100% but didn’t try making a BackEnd object yet and do several tests since was getting late.