**Assignment 6:**

**1. Explain differences in flow control js and other languages like python, c with example code.**

Basically node is single threaded as it uses javascript (which is single threaded) as its programming language.

It is able to perform multiple tasks at a time under some set of rules like event loop and promise.

It contains multiple levels of phases to handle different type of delays.

For example: Reading a file in node is not a synchronous process as the request is sent to the poll phase. In the meantime other processes will take place. Once the response from the file is received then the file processing will take place. This is a feature of the event loop.

While other languages like c,c++, etc are not able to perform multiple tasks efficiently.

Each step in them is executed synchronously.

For example: Reading a file takes a little while in c, the entire process will wait for the file response.

**2. Explain nodejs event loop.**

**EVENT LOOP:**

┌───────────────────────────┐

┌─>│ timers │

│ └─────────────┬─────────────┘

│ ┌─────────────┴─────────────┐

│ │ pending callbacks │

│ └─────────────┬─────────────┘

│ ┌─────────────┴─────────────┐

│ │ idle, prepare │

│ └─────────────┬─────────────┘ ┌───────────────┐

│ ┌─────────────┴─────────────┐ │ incoming: │

│ │ poll │<─────-┤ connections, │

│ └─────────────┬─────────────┘ │ data, etc. │

│ ┌─────────────┴─────────────┐ └───────────────┘

│ │ check │

│ └─────────────┬─────────────┘

│ ┌─────────────┴─────────────┐

└──┤ close callbacks │

└───────────────────────────┘

**Timer phase:**

It is the first phase to work in an event loop.

It contains a message queue which contains all the timeout events in ascending order. If the time reachout the threshold limit the task will start to execute.

**Pending callback phase:**

It works after the pending timer phase.

It contains the events that will all need to be immediately executed. The setImmediate() methods are queued here.

**Idle:**

This phase is used to perform background tasks such as garbage collection, checking for low priority events..

idle.ignore() is used to avoid the execution of the idle phase.

**Poll:**

It works after the pending callback phase.

It handles all the I/O requests. If a i/o request is made or canceled due to some error then the process will be made. Meanwhile the other synchronous tasks take place.

**Check phase;**

It works after the Poll phase.

If any setImmediate occurs after the pending callback phase it is handled in this phase.it doesn't wait for the next event loop cycle.

**Close callback phase;**

It works after the check phase.

It contains events which are handled during the closing time, that is while closing a socket.

* Before starting an event loop it checks if there are any further tasks available. If any the cycle starts, if not then it is set to idle state or to the end state.