Queue Program

//allow dynamic user input for adding/removing doc //stack and queue behaviors

-enqueue -dequeue -reading input -assigning input (check for case) -user ability to: -print -view -add -remove -quit

stack: container of objects that follow "Last in, first out" principle queue: container of objects that follow "First in, first out" principle

A queue structure makes the most senese for a printing simulation because it follows the "First in, first out" principle. This means that when a document is added to the queue, it will be placed at the back of the line. This ensures priority. If a stack had been implemented, a document added to the structure would immediately print that item before anything else.

Functions: AddDoc, PrintDoc, ViewDoc, RemoveDoc

static void AddDoc() { console.WL("Enter name of doc: "); string doc = console.RL(); printQueue.Enqueue(doc); console.WL("Doc '{doc}' added); }

static void PrintDoc() { if (printQueue.count > 0) { string printedDoc = printQueue.Dequeue(); console.WL("Printing '{printedDoc}'); } }

static void ViewQueue() { console.WL("Current print queue: "); if (printQueue.count > 0) { foreach(string doc in printQueue){ console.WL("-{doc}"); } } else { console.WL("print queue is empty"); } }