ITP 365

Lecture 4 – 1/19/2017

Stacks and Queues

* **STRINGS ARE COLLECTIONS OF CHARS**
* Vector example – Average
  + Pass by reference because more efficient even if avg. doesn’t change vector
* Queue – collection
  + Allows to add elements to end (tail) and remove from front (head)
  + **Cannot read from any arbitrary element unlike arrays**
  + FIFO (first in, first out)
  + Ex. Printer
    - Queue to a movie
    - Can only access beginning of a queue cant access middle
  + #include “queue.h”
  + Queue<int> myQueue;
* Queue member functions
  + isEmpty - returns bool
  + clear – clears everything from queue
  + size – returns number of elements in queue
  + **Enqueue** – adds an element to the tail (end)
    - myQueue.enqueue(element);
  + **Dequeue** – removes element at the head (front) of queue – **cannot read from the tail**
    - Cout << myQueue.dequeue() << endl;
  + **Peek** – returns element at the head (front) but does NOT remove item
* Stacks – collection
  + Allows you to add elements to top of stack
  + Allows you to remove elements from the top of the stack
  + **Cannot read from any arbitrary element unlike arrays**
  + LIFO (last in, first out)
  + Ex. Stack of plates at parkside
    - the last plate put on stack is first to be washed
  + #include “stack.h”
  + Stack<int> myStack;
* Stack member functions
  + Push – push an element onto top of the stack
    - myStack.push(element)
  + pop – removes element form the top of the stack and returns it
    - type OldTop = mystack.pop();
* **Infix notation – binary operators appear in between operands**
  + **5 + 10**
  + **not efficient way for a computer to perform arithmetic with numbers**
* **Postfix notation – appears after operands**
  + **5 10 +**
  + **5 10 + 2 \* means (5 + 10) \* 2**
* **Physical calculator examples**