**6/4/2013 – Introduction to C Data Structures**

Data structures

* Designed to be accessed efficiently
* Composite of related items
* Main Types: Linked lists, Stacks, Queues, Binary Search Trees (BST)
  + Binary search tree is not linear
* An organization of information for better algorithm efficiency. It may include redundant information such as length of the list or number of nodes in a subtree

Abstract Data type (ADT)

* A data structure is one implementation of an abstract data type
* A set of data values and associated operations that are precisely specified independent of any particular implementation
  + Data values, operations, how they are specified

Linked Lists

* Collections of data items lined up in a row
* Insertions/deletions may be made anywhere

Stacks

* Restricted linked lists
* Insertions/Deletions may be made at one end only

Queues

* Insertions at the back and deletions at the front

Binary Search Trees

* Require linked data items
* Efficient for searching and sorting of data

Assignment 3:

* 12.2 infix to postfix (using stack)
* 12.3 evaluate postfix (using stack)

C++ Classes:

* Classes -> Objects
* Public, Private, Protected
* Stream insertion, stream extraction
* Function overloading, operator overloading
* This: Refers to an instance of an object
* Member vs Non-member function
* Pass by reference vs pass by value
* Constructor vs copy constructor
* Destructor
* Getters(accessors) & Setters(mutators)

PA6

1. Fetch (grab an instruction from memory <array>)
   1. Look out for instruction counter (integer) which is an index in memory array
2. Decode
3. Execute

UML = unified modelling language

* Class diagram
  + “+” public members +setPayRate(in newPayRate : double) : void
  + “-“ private members -mlsASalariedEmployee : bool
  + “#” protected members #mPayRate: double

Overridden function: Virtual (has the same name & same parameters)

Overloaded function: has the same name (but different parameters) – Do not use virtual

Final Exam

* Encapsulation
* Information hiding
* Function overloading
* Access specifier