**Bronx Community College**
**Department of Mathematics and Computer Science**
**Fall 2019 CSI33**
**Review for Midterm Exam**

## Chapter 1: Abstraction and Analysis

**Function specification** tells what a function does but not how that is done. The function **precondition** tells what is true before the function is called. The function **postcondition** tells what is true after the function finishes executing—the significance of the return value or the effect of a function on some argument. Functions should not produce undocumented side effects.
Enforcing preconditions: raise an exception or use an assert statement
Analysis of algorithms: how many steps are needed to carry out an algorithm? how does the number of steps needed to carry out the algorithm increase as the size of the problem increases?
Big O bounds- Big O provides an upper bound
Big Theta bounds- Big Theta provides both an upper and a lower bound
Estimating time to solve problems

## Chapter 2 Data Abstraction

Abstract Data type (ADT) - type (or class) for objects with behavior defined by a set of values and a set of operations. ADT tells what operations are to be performed, but not how operations will be implemented, does not specify how data will be represented or and what algorithms will be used for operations
ADT examples: Dataset, Rational, Card
**Class invariant** is a set of properties that all objects in the class must satisfy. It gives a criterion for the object to be valid, must be fulfilled by the constructor, must be maintained by the public methods.
Unit-testing using Python’s unittest module, TestCase class, assertEqual methods

## Chapter 3: Container Classes

Python list methods
Deck of Cards, Hand of Cards using Python list container
Overloading comparison operators for Card objects, Python hook methods
Array data collection, Python list implementation by dynamic arrays of references
implementation of list operations, time efficiency of indexing, insert, append, delete, other operations
Python dictionary ADT, implementation using hash table, hashing function

## Chapter 4 Linked Structures and Iterators

Arrays: homogeneous, size determined statically unless additional programming, efficient random memory access
Python memory model, references to objects, namespace dictionary, aliasing, shallow copy, deep copy, parameter passing
Linked lists using List Node and links, head reference, instance variable, size instance variable.
Implementation of methods needed to make a linked list function as a Python list – indexing, insert, delete, append, pop, others
Implementation of Python iterator
Time analysis of various operations
Programming exercise: adding tail reference to linked list, modifying method definitions to preserve class invariant
Time analysis of linked list operations with tail reference

## Chapter 5 Stacks and Queues

**Stack ADT:** last in, first out ADT, methods - push to place data, pop to remove data, top to examine data
Stack implemented with Python list, with linked list
Stack applications: evaluating Reverse Polish expressions, changing infix expressions to reverse Polish expressions
**Queue ADT:** first in, first out ADT, methods – enqueue to add data, dequeue to remove and return data, front to examine data
Queue implementation using python list, linked list
Queue applications in operating systems, simulations

## Chapter 6 Recursion

Recursive function – a function can be defined using the function itself in the definition. The definition must have one or more base cases where the function can be computed without a recursive call. Every chain of recursive calls must terminate at a base case.

Examples – factorial, string reversal, permutations, exponentiations by repeated squaring, binary search
Time analysis of recursive algorithms, comparison to other algorithms using loops
Recursive sorting algorithm – mergesort. Time and space analysis of mergesort.
Recursive solution of Tower of Hanoi puzzle. Time analysis of Tower of Hanoi

**Exam structure:**
Part I: paper and pencil, closed notes and book, no use of computer, multiple choice, true-false, short answer
Covering definitions and concepts, programming techniques
Part II: open book, open notes, using computer, programming exercises