SOLUTION KEY

CSCI 230 EXAM 2 - Closed book/notes/neighbor............

1) 10 pts. Write a method that returns ***true*** if two 1D arrays of ints are identical. For example, **same({6,4,2,9}, {6,4,2,9})** returns true but **same({1,2,3,4,5,6,7,8}, {1,2,3})** returns false.

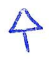
```
public boolean same(int a[ ], int b[ ]){
    if (a.length() != b.length()){
        return false;
    }else {
        for(int i=0; i < a.length(); ++i){
            if( a[i] != b[i]){
                return false;
            }
        }
    }
    return true;
}
```

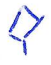
2) 10 pts. Fill in the following table:

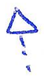
Search Strategy	Time Big-O	Space Big-O	Complete	Optimal
Depth-first	b^d	$b \cdot d$	N	N
Breadth-first	b^d	b^d	Y	Y
Depth-limited	$b^{d==L}$	$b \cdot d==L$	Y if d<L	N
Best-first	b^d	b^d	N	N
A*	b^d	b^d	Y	Y

if heuristic is admissable: it doesn't overshoot the goal.

3) 5 pts each. Draw the UML symbol/notation for:

Inheritance relationship 

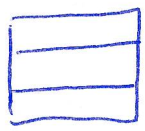
Composition relationship 

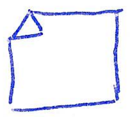
Implementing an Interface 

public +

private −

protected #

Class Diagram 

Comment 

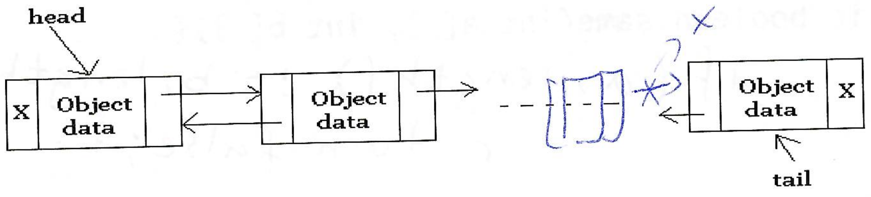
4) 5 pts. A class that implements the built-in Java Comparable interface must do what?

It must implement the compareTo() method which returns an int: 0 = same; negative/positive depending on order.

5) 5pts. "Object" is a special object in Java. Explain why.

Sits on top the Java hierarchy. Every object in Java (user built or pre-defined) is-a Object.

6) 20pts. Recall that we covered the LinkedList data type in class in which each LinkedListNode has a single reference to the 'next' node in the list. A "double linked list" data type is like a linked list in which each node has two references: one reference points to the 'next' item and the other points to the 'previous' item. This is illustrated in the below picture. Write two methods: removeBack() which removes and returns the last item in a double linked list, and printReverse() which simply prints each item in the linked list in reverse order.

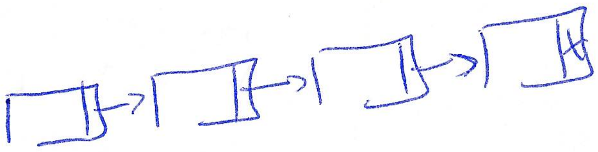


```
public class dNode {
      private Object data;
      private dNode next, prev;
      …get/set methods…
      …etc…
}

public class doubleLinkedList {
      private dNode head, tail;

public Object removeBack(){
// This is worth 10 points
    if (this.head == null) {
            return null;
    }else if(this.head.getNext() == null) {
            Object temp = this.head.getData();
            this.head = null;
            this.tail = null;
            return temp;
    }else {
            Object temp = this.tail.getData();
            this.tail = this.tail.getPrev();
            this.tail.setNext(null);
            return temp;
    }
}
public void printReverse( ){
// This is worth 10 points
      dNode temp = this.tail;
      while(temp != null) {
            System.out.println(temp.getData());

      }

}
```

7) 30 pts. Recall our LinkedList and Node objects we wrote in class:

```
public class Node{
        private Object data;
        private Node next;
        ...etc...
        ...etc...
}
public class LinkedList{
        private Node head;
        ...etc...
        ...etc...
}
```

Write these new methods for the LinkedList class:



```
public boolean lengthAtLeast(int len){
// This is worth 10 points.
// Returns true if there are at least len number of Nodes in the LinkedList.
    Node temp = this.head;
    for(int i = Ø; i < len; ++i) {
            if (temp == null){
                    return false;
            }
            temp = temp.setNext();
    }
    return true;
}

public LinkedList reverse(){
// This is worth 10 points.
// Returns a LinkedList with the nodes in reverse order.
        LinkedList ret = new LinkedList();
        Node temp = this.head;
        while(temp != null){
                if(ret.setHead() == null){
                        ret.setHead(temp);
                }else{
                        Node newNode = new Node(temp.setData());
                        newNode.setNext(ret.setHead());
                        ret.setHead(newNode);
                }
        }
}
```

```
public boolean insertAtPosition(Object item, int index){
// This is worth 10 points
// The head of the LinkedList points to item at index 0, the second item is at index 1, etc.
// This method inserts a new item at the index location. If an existing item is already
// at this index, move it back to location index+1. If the index is equal to the number of items
// in the LinkedList, then the new item gets inserted at the end of the linked list.
// If the index is less than 0 or greater than the number of items in the LinkedList, this method
// returns false. Otherwise, this method returns true after correctly inserting the item.
    LinkedListNode    newNode = new LinkedListNode
                                                (item)
    if (index < 0) {
            return false;
    }else ~~if~~ (index == 0);
        newNode.setNext(this.head);
        this.head = newNode;
        return true;
    } else if (this.head == null) {
        return false;
    }else{
        LinkedListNode temp = this.head;
        for(int i =0; i < index-1; ++i) {
                if(temp.getNext() == null) {
                        return false;
                }
                temp = temp.getNext();
        }
        newNode.setNext(temp.getNext());
        temp.setNext(newNode);
        return true
    }
}
```

8) 5pts. What is the difference between a Stack data structure and a Queue data structure?

stack = FILO queue = FIFO

9) 5pts. What is the difference between ASCII and UNICODE?

ASCII = 1 byte
UNICODE = 2 bytes } both encode alpha/numeric characters + symbols to binary

10) 5pts. In Object Oriented Programming, what does a constructor method do? How many can you have? What's the name of the constructor method?

initialize the object data. Any number as long as the arguments are different. The name is same as class name.