

SAMPLE EXAM
First Exam
Computer Programming 338
Dr. St. John
Lehman College
City University of New York
Tuesday, 8 March 2011

NAME (Printed) _____
NAME (Signed) _____
E-mail _____

Exam Rules

- Show all your work. Your grade will be based on the work shown.
- The exam is closed book and closed notes.
- When taking the exam, you may have with you pens or pencils, and an 8 1/2" x 11" piece of paper filled with notes, programs, etc.
- You may not use a computer or calculator.
- All books and bags must be left at the front of the classroom during this exam.
- All pseudocode is from <http://en.wikipedia.org/wiki/> unless otherwise noted.
- **Do not open this exams until instructed to do so.**

Question 1	
Question 2	
Question 3	
Question 4	
Question 5	
Question 6	
Question 7	
Question 8	
Question 9	
Question 10	
TOTAL	

1. True or False:

- (a) ___ $n^4 + 12n^2 = O(n^4)$.
- (b) ___ $n^2 = O(n^2)$.
- (c) ___ $\lg n = O(n)$.
- (d) ___ For all functions, f_1, f_2, g_1, g_2 , $f_1 = O(g_1)$ and $f_2 = O(g_2)$ implies $f_1 + f_2 = O(g_1 + g_2)$.
- (e) ___ For all functions, f, g , $f = O(g)$ implies $2f = O(g)$.

2. Given data that is nearly sorted (that is, most of the inputted data is in increasing order), which of the sorts below would have the fastest running time? Justify your answer:

```

procedure selectionSort( A[] )
  for each iPos in 0 to A.length-1
    iMin = iPos;
    for (i = iPos+1; i < n; i++)
      if (a[i] < a[iMin])
        iMin = i;
    if ( iMin != iPos )
      swap(a, iPos, iMin);
end procedure

```

```

procedure bubbleSort( A[] )
  do
    swapped = false
    for each i in 1 to length(A)-1:
      if A[i-1] > A[i] then
        swap( A[i-1], A[i] )
        swapped = true
    while swapped
end procedure

```

3. Below is a table of two sorts run on lists of numbers in increasing, decreasing, and random order.

	Increasing				Decreasing				Random			
	100	200	300	400	100	200	300	400	100	200	300	400
$sort_1$	0	3	5	6	1	8	10	14	1	10	15	18
$sort_2$	0	0	1	2	0	0	2	4	0	1	3	5

- (a) Which sort is better on increasing data? Why?
- (b) Which sort is better overall?
- (c) From your table can you conclude that $sort_1 = O(sort_2)$? Why or why not?

4. What is the running time of the following pieces of code? To get full credit, analyze the code line by line and justify your answer.

(a) `String s = "exam 1";`
`LinkedList<String> names = new LinkedList<String>();`
`names.add(s);`

(b) `procedure gnomeSort(a[])`
 `pos := 1`
 `while pos < length(a)`
 `if (a[pos] >= a[pos-1])`
 `pos := pos + 1`
 `else`
 `swap a[pos] and a[pos-1]`
 `if (pos > 1)`
 `pos := pos - 1`
 `else`
 `pos := pos + 1`
 `end if`
 `end if`
 `end while`

5. Write a method that as input a Queue and removes that subsequent elements in the list that are identical (i.e. removes identical elements). Use the Queue API below:

```
java.util Interface Queue<E>
boolean add(E e)    Inserts the specified element into this queue if capacity,
                    returning true upon success and throwing an IllegalStateException
                    if no space is available.
E element()         Retrieves, but does not remove, the head of this queue.
boolean offer(E e)  Inserts the specified element into this queue if it is possible
                    to do so immediately without violating capacity restrictions.
E peek()            Retrieves, but does not remove, the head of this queue,
                    or returns null if this queue is empty.
E poll()            Retrieves and removes the head of this queue, or returns
                    null if this queue is empty.
E remove()          Retrieves and removes the head of this queue.
```

6. Using the “By Hand” implementation of a linked list

```
public class myList {                public class Elem {
    Elem front;                      String data;
    Elem back;                      Elem next;
    myList() {front=back=null;}      Elem(String s) {data=s; next=null}
}
```


(a) Write code that will declare a linked list, called `FirstNames` to hold strings:

(b) Write a method that will concatenate two linked lists, and return your new list:

7. Using the classes above, draw a picture of memory after each of the following blocks of code are executed:


MyList line = new MyList();
(a) line.front = new Elem("blue");
line.back = line.front;

front back




Elem t = new Elem("yellow");
(b) line.back.next = t;

front back



Elem t = new Elem("green");
(c) t.next = line.front.next;
line.front.next = t;

front back



8. Given a queue implemented as a circular array, indicate the state of the variables after each block of code is executed:

```
int[] line = new line[10];  int front = 0, back = 0;
```

(a) `line[back++] = -5;`
`front++;`

line									
0	1	2	3	4	5	6	7	8	9

(b) `for (int i = 0; i < 10; i++)`
`{ line[back] = i;`
`back = (back+1)%10;}`

line									
0	1	2	3	4	5	6	7	8	9

(c) `front++;`

line									
0	1	2	3	4	5	6	7	8	9

9. What is the running time of the following `radixSort`, in terms of n , the length of the list, and k the number of unique items in the list. To get full credit, analyze the code line by line and justify your answer.

```
public static void radixSort(int array[], int digits)
{ for (int i = digits-1; i >= 0; i--)
    insertionSort(array, i);
}
public static void InsertionSort(String[] A, int keyIndex){
    for (int j =1; j < A.length; j++)
    { String key = A[j];
      int i = j-1;
      while ((i >= 0)&&(A[i].charAt(keyIndex)>key.charAt(keyIndex))) {
          A[i+1] = A[i];
          i = i-1;
      }
      A[i+1] = key;
    }
}
```

10. (a) Write a method that takes a queue of customers (implemented as a linked list of customers) and the current time, and removes the first element of the queue if its arrival time is after the current time. You may assume that the class `Customer` contains a method:

```
public static int arrivalTime()  
that returns the arrival time for that customer.
```

- (b) Assume there are 5 check-out counters and array of `boolean`s that indicate if the counter is busy:

```
boolean[] isBusy = new boolean[5];
```

Write a new method that inserts the customer into the first line that is not busy: