THE UNIVERSITY OF AUCKLAND

SUMMER SEMESTER, 2007
Campus: City

COMPUTER SCIENCE

Principles of Computer Science

(Time Allowed: ONE hour)

NOTE: You must answer **all** questions in this test.
No calculators are permitted
Write your answers in the space provided.
There is space at the back for answers that overflow the allotted space.
The test is out of 85 marks

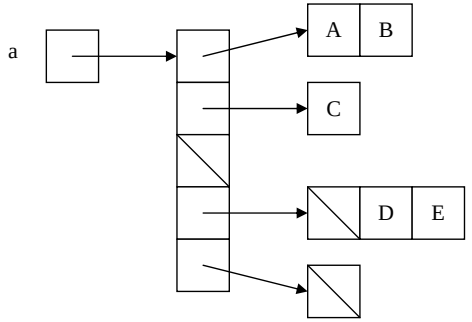
Surname	
Forenames	
Student ID	
Login (UPI)	

	Question	Mark	Out Of
1	Multi-dimensional Arrays		10
2	Exceptions		10
3	Recursion		10
4	Sorting		10
5	Performance Analysis		10
6	Abstract Data Types		15
7	Linked Lists		10
8	Stacks		10
	TOTAL		85

ID..…….

1. Multi-dimensional Arrays (10 marks)

The following diagram shows the contents of a two-dimensional array of Strings. Write the Java code that would be used to create an array with the structure and contents shown by the diagram.



Write your Java code in the space below:

```
String[][] a = {{"A","B"}, {"C"}, null, {null,"D","E"}, {null}};

or

String[][] a = new String[5][];
a[0] = new String[2];
a[0][0] = "A";
a[0][1] = "B";

a[1] = new String[1];
a[1][0] = "C";

a[2] = null;

a[3] = new String[3];
a[3][0] = null;
a[3][1] = "D";
a[3][2] = "E";

a[4] = new String[1];
a[4][0] = null;
```

(10 marks)

2. Exceptions (10 marks)

Write a method called `readPercentage()` that will continually ask the user to enter a number between 1 and 100 until the input entered by the user is a valid integer number. If the user enters any text except an integer number between 1 and 100, then the program should ignore the text and ask the user again.

An example of the output produced by executing the `ReadValidInput` application is shown below. Note that the user input is displayed in bold typeface.

```
C:>java ReadValidInput
Please enter a number between 1 and 100:  -23
Please enter a number between 1 and 100:  3.5
Please enter a number between 1 and 100:  thirteen
Please enter a number between 1 and 100:  13
Thank you.  You entered the number 13
```

The application `ReadValidInput` will make use of the `readPercentage()` method as follows:

```
public static void main(String[] args) {
   int input = readPercentage();
   System.out.println("Thank you.  You entered the number " + input);
}
```

Your method may make use of the `Keyboard` class to simplify reading input from the user. The `Keyboard` class contains the following method:

```
public static String readInput()    //reads a String from the user
```

Write your Java code in the space below:

```
private static int readPercentage() {

    int input = -1;
    while(input < 1 || input > 100) {
       try{
          System.out.print("Please enter a number between 1 and 100:  ");
          input = Integer.parseInt(readInput());
       } catch (NumberFormatException e) {}
    }
    return input;

}
```

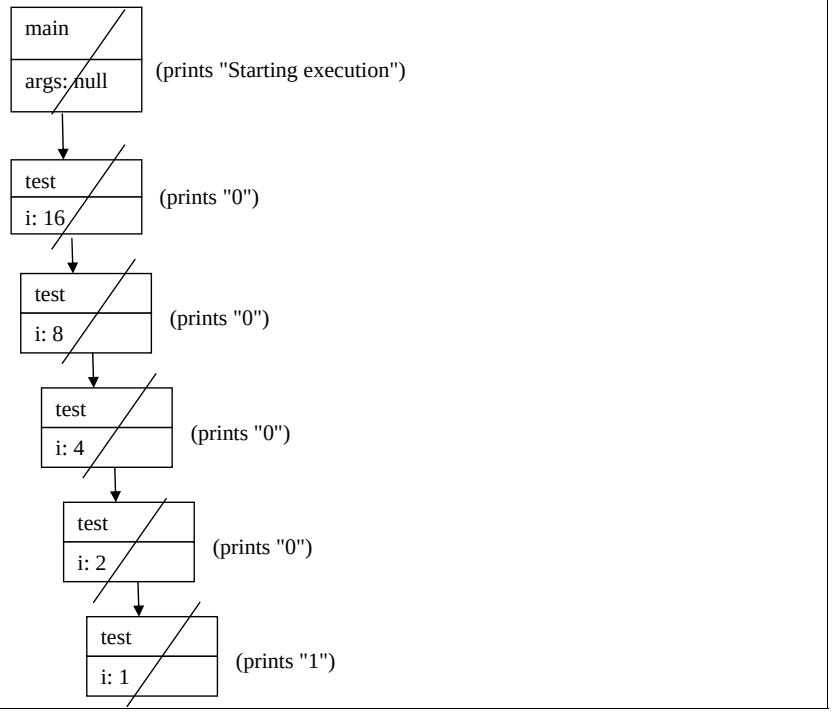
(10 marks)

3. Recursion (10 marks)

(a) Perform a code trace (called a box trace by the textbook) of the following code.

```
public class TestRecursion {
   public static void main(String[] args) {
      System.out.println("Starting execution");
      test(16);
   }
   private static void test(int i) {
      if (i < 2)
         System.out.print(i);
      else {
         System.out.print(i%2);
         test(i/2);
      }
   }
}
```

Show the value of all parameters. Also indicate when output is produced by a method call.



(5 marks)

(b) The following code differs slightly from the previous method. What is the output produced when the code is executed?

```
public class TestRecursion {

   public static void main(String[] args) {
      test(19);
   }

   private static void test(int i) {
      if (i < 2)
         System.out.print(i);
      else {
         test(i/2);
         System.out.print(i%2);
      }
   }
}
```

Output:

```
10011
```

(5 marks)

4. Sorting (10 marks)

(a) The following code performs a sorting algorithm. Note that the swap method performs the task of swapping two elements of the array as described in class.

```
public void sort(int[] data) {
   for (int i = 0; i < data.length - 1; i++){
      int index = i;
      for (int j = i + 1; j < data.length; j++)
         if (data[j] < data[index])
            index = j;
      swap(data, i, index);
   }
}
```

(i) Name the sorting algorithm shown by the code above:

```
Selection sort
```

(2 marks)

(ii) Explain how this version of the sorting algorithm goes about sorting the elements in the array.

```
The algorithm finds the smallest element and swaps it with the element
in the left-most location.  The next smallest is swapped with the next
element and so on.  In each iteration, the smallest element of the
unsorted elements is located and swapped into the appropriate position.
```

(3 marks)

(b) The following code is used to partition the array during a quicksort.

```
private int partition(int[] theArray, int first, int last) {
   int pivot = theArray[first];
   int lastS1 = first;

   for (int firstUnknown = first + 1; firstUnknown <= last;
                                                   firstUnknown++) {
      if (theArray[firstUnknown] < pivot) {
         lastS1++;
         swap(theArray, firstUnknown, lastS1);
      }
      //SHOW THE ARRAY AT THIS POINT
   }

   swap(theArray, first, lastS1);
   //SHOW THE ARRAY AT THIS POINT
   return lastS1;
}
```

Assume that the partition method has been called with the following method call:

```
partition(a, 2, 7);
```

Given the initial state of the array a as indicated below, show the contents of the array at the points indicated in the code above.

a

4	1	6	8	2	5	7	9	3

4	1	6	8	2	5	7	9	3

4	1	6	2	8	5	7	9	3

4	1	6	2	5	8	7	9	3

4	1	6	2	5	8	7	9	3

4	1	6	2	5	8	7	9	3

4	1	5	2	6	8	7	9	3

(5 marks)

5. Performance Analysis (10 marks)

(a) State the big-O of the following functions:

(i) $3n^2 + n * \log_2 n + 50n$

$O(n^2)$

(2 marks)

(ii) 100

$O(1)$

(2 marks)

(iii) $2^n + n^2$

$O(2^n)$

(2 marks)

(b) What is the big-O of the following algorithms:

(i)
```
private static void foo(String[] a) {
  for (int i = 0; i < a.length; i = i + 2) {
    System.out.println(i);
  }
}
```

`O(n)`

(2 marks)

(ii)
```
private static void foo2(String[] a) {
  for (int i = 1; i < a.length; i = i * 2) {
    System.out.println(i);
  }
}
```

`O(logn)`

(2 marks)

6. Abstract Data Types (15 marks)

(a) Here is the Java interface to an Abstract Data Type.

```
public interface DoubledIntList {

   /*
    * Returns the number of integers in the list.
    */
   public int size();

   /*
    * Adds the element on the end of the list.
    */
   public void add(int element);

   /*
    * Inserts the element at the index position of the list.
    */
   public void set(int index, int element);

   /*
    * Returns twice the element originally entered at the index position.
    */
   public int get(int index);

}
```

This is an integer list with a difference. All integers retrieved from the list using `get` are double the value that was entered into the list. For example, the following code which uses this interface produces the output shown.

```
public class UseDoubledIntList {

   public static void main(String[] args) {
      DoubledIntList dil = new DILImplementation();
      dil.add(12); dil.add(7); dil.add(3);
      dil.set(2, 19);
      for (int i = 0; i < dil.size(); ++i)
         System.out.println(dil.get(i));
   }
}
```

Produces:

```
24
14
38
```

Notes:

- The first element of the list is at index 0.
- You do not have to worry about errors, such as non-existing elements or adding too many elements.

Complete the `DILImplementation` class which implements the `DoubledIntList` interface to give the result above.

```
public class DILImplementation implements DoubledIntList {
   private int size = 0;
   private int[] data = new int[10];

   public void add(int element) {
     data[size++] = element * 2;
   }

   public int get(int index) {
     return data[index];
   }

   public void set(int index, int element) {
     data[index] = element * 2;
   }

   public int size() {
     return size;
   }
}
```

(10 marks)

(b) Describe an alternative implementation of the `DoubleIntList` which would give the same results.

In my case, rather than doubling the value when inserting it into the array, I could have doubled it in the get method. A completely different answer such as using a linked implementation is also acceptable.

(5 marks)

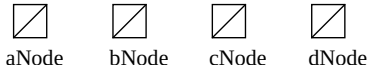
7. Linked Lists (10 marks)

Here is the Node class from lecture 14.

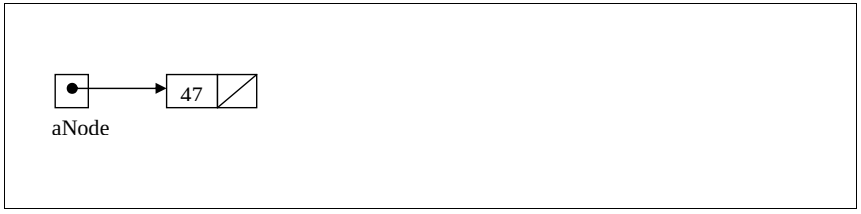
```
public class Node {

   private Object item;
   private Node next;

   public Node(Object item, Node next) {
      this.item = item;
      this.next = next;
   }

   public Object getItem() {
      return item;
   }

   public void setItem(Object item) {
      this.item = item;
   }

   public Node getNext() {
      return next;
   }

   public void setNext(Node next) {
      this.next = next;
   }

}
```

Complete the node diagrams as in lectures to show the results of the following statements. For each statement you must show the item and next fields for all existing Node objects.

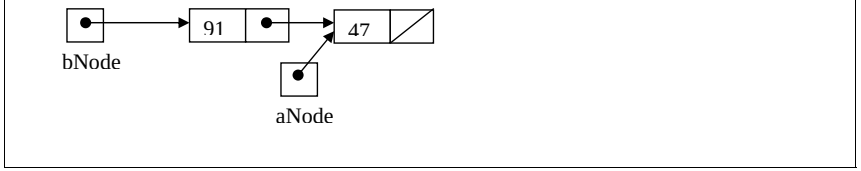
```
Node aNode, bNode, cNode, dNode; // assume they are all null
```



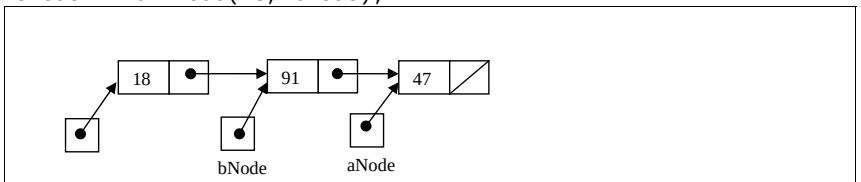

```
aNode = new Node(47, null);
```



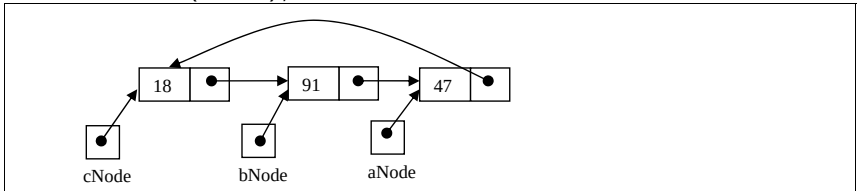

```
bNode = new Node(91, aNode);
```



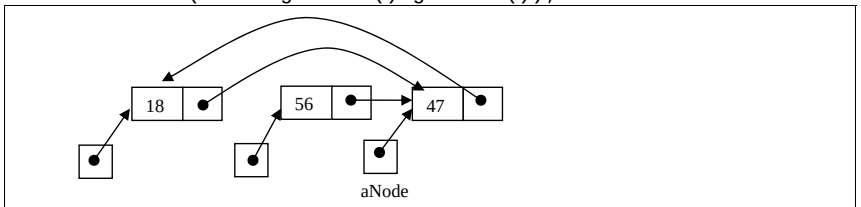

```
cNode = new Node(18, bNode);
```




```
aNode.setNext(cNode);
```




```
cNode.setNext(cNode.getNext().getNext());
```



(10 marks)

8. Stacks (10 marks)

(a) For each of the following strings, carry out the balanced-braces algorithm and show the contents of the stack at each step.

(i) `a{b{c{d}}`

```
a Stack:
{ Stack: {
b Stack: {
{ Stack: { {
c Stack: { {
{ Stack: { { {
d Stack: { { {
} Stack: { {
} Stack: {
```

Result

```
Not balanced
```

(ii) `ab{c}d{{ef}g{h}}`

```
a Stack:
b Stack:
{ Stack: {
c Stack: {
} Stack:
d Stack:
{ Stack: {
{ Stack: { {
e Stack: { {
f Stack: { {
} Stack: {
g Stack: {
{ Stack: { {
h Stack: { {
} Stack: {
} Stack:
```

Result

```
balanced
```

(10 marks)

CONTINUED

- Overflow Sheet 1 -
Write the question number and letter next to your answer. You must ALSO indicate in the allotted space that you have used the overflow sheet.

CONTINUED

ID..……

- Overflow Sheet 2 -
Write the question number and letter next to your answer. You must ALSO indicate in the allotted space that you have used the overflow sheet.

CONTINUED

ID..……

- Overflow Sheet 3 -
Write the question number and letter next to your answer. You must ALSO indicate in the allotted space that you have used the overflow sheet.

CONTINUED

ID..……

Rough Working – This page will not be marked

ID..……

Rough Working – This page will not be marked