University of New England

UNIVERSITY OF NEW ENGLAND

NAME:____________________________________

STUDENT NUMBER:_________________________

UNIT NAME: COMP132

PAPER TITLE: Computer Science II

PAPER NUMBER: First and Only

DATE: **Monday 21 November 2011** **TIME:** **9:15 AM TO 11:30 AM**

TIME ALLOWED: TWO (2) hours and fifteen minutes

NUMBER OF PAGES IN PAPER: FIFTEEN (15)

NUMBER OF QUESTIONS ON PAPER: NINE (9)

NUMBER OF QUESTIONS TO BE ANSWERED: NINE (9)

STATIONERY PER CANDIDATE:	**0**	6 LEAF A4 BOOKS	**1**	ROUGH WORK BOOK	**0**	GENERAL PURPOSE ANSWER SHEET
	1	12 LEAF A4 BOOKS	**0**	GRAPH PAPER SHEETS	**0**	SEE OTHER 'AIDS REQUIRED' BELOW

OTHER AIDS REQUIRED: NIL

POCKET CALCULATORS PERMITTED: NO

TEXTBOOKS OR NOTES PERMITTED: NIL

INSTRUCTIONS FOR CANDIDATES:

- Candidates MAY NOT start writing until instructed to do so by the supervisor
- Please pay attention to the announcements and read all instructions carefully before commencing the paper
- Candidates MUST write their name and student number on the top of this page
- Questions are NOT of equal value. Marks are allocated at the beginning of each question
- **Question 1:** – Multiple choice – Answer ALL questions by circling the letter that corresponds with the chosen answer on this examination paper
- **Questions 2 to 9:** – Answer ALL questions in the answer booklet provided
- This examination question paper **MUST BE HANDED IN** with worked scripts. Failure to do so may result in the cancellation of all marks for this examination

REMEMBER TO WRITE YOUR NAME AND STUDENT NUMBER AT THE TOP OF THIS PAGE

THE UNIVERSITY CONSIDERS IMPROPER CONDUCT IN EXAMINATIONS TO BE A SERIOUS OFFENCE. PENALTIES FOR CHEATING ARE EXCLUSION FROM THE UNIVERSITY FOR ONE YEAR AND/OR CANCELLATION OF ANY CREDIT RECEIVED IN THE EXAMINATION FOR THAT UNIT.

Question 1

Multiple Choice: Choose the one alternative that best completes the statement or answers the question.

(a) *[2 marks]*

Given the following code:

```
public static int myTest(int x, int y)
{
  if ( x > y)
  {
    return 10;
  }
  else
  {
    return (myTest(x + 5, y - 5) + 5);
  }
}
```

What value is returned for the following method call:

```
System.out.println(myTest(9,13));
```

(i) 10

(ii) -5

(iii) 15

(iv) Infinite loop

(b) *[2 marks]*

Which of the following does the JVM not have to perform when a method is called?

(i) Allocate memory for local variables

Question 1 (b) *(ii)* **is on page 3**

(ii) Allocate memory for parameters

(iii) Terminate the calling program

(iv) Store the address of the program location where to return after the method terminates

(c) *[2 marks]*

The two criteria most often used to measure the efficiency of an algorithm are:

(i) time and method overhead

(ii) space and time

(iii) execution time and speed

(iv) the number of iterations

(d) *[2 marks]*

For an unsorted list, the best algorithm to use to search for a value is:

(i) bubble search

(ii) sequential search

(iii) insertion search

(iv) binary search

Question 1 (e) is on page 4

(e) *[2 marks]*

The class

```
public class Point3D<T extends Number> extends Point<T>
{
  ...
}
```

is an example of:

(i) a generic class extending the class Number

(ii) a non-generic class extending a generic class

(iii) a generic class extending a non-generic class

(iv) a generic class extending another generic class

(f) *[2 marks]*

Given the following method header:

```
void displayPoint(Point<? super Double> myPoint)
```

Which of the following objects can be passed as an argument to the displayPoint method?

(i) `Point<Double> p;`

(ii) `Point<Number> p;`

(iii) `Point<Object> p;`

(iv) All of the above

Question 1 (g) is on page 5

(g) *[2 marks]*

A class is generic:

(i) if it is a superclass of the Object class

(ii) if it has type parameters

(iii) if it is a subclass of the Object class

(iv) if it has method parameters

(h) *[2 marks]*

Which of the following is true?

(i) The load factor of Tree Set should never exceed 50% to ensure maximum performance

(ii) A TreeSet created with the no-arg constructor expects its elements to implement the Comparable interface

(iii) A TreeSet created with the no-arg constructor expects its elements to implement the Comparator interface

(iv) The load factor of Tree Set should never exceed 75% to ensure maximum performance

(i) *[2 marks]*

Which of the following is true?

(i) Collections that implement the Collection interface do not support generic types

(ii) Collections that implement the Map interface do not support generic types

(iii) Collections that implement the SortedMap interface do not support generic types

(iv) All of the collections in the JCF support generic types

Question 1 (j) is on page 6

(j) *[2 marks]*

If a new element is added to an ArrayList whose internal array if already full,

(i) the new element is not added, and null is returned

(ii) the add method throws an exception

(iii) A new, bigger internal array is created, the elements are moved to the new array, the old internal array is deleted, and then the new element is added

(iv) the new element is not added, and -1 is returned

(k) *[2 marks]*

When using recursion on linked lists

(i) the recursive method should be one of the methods specified in the List interface

(ii) the recursive method should not call itself outside of its base case

(iii) the recursive method should be made private, and should be called by a public non-recursive method

(iv) the linked list class is subclassed, and then the recursive method overrides a method of the same name in the list class

(l) *[2 marks]*

A **circularly linked** list makes it easy to

(i) move from any node to its predecessor

(ii) jump from the last node to the first

(iii) jump from node to node

(iv) move from any node to its successor

Question 1 (m) is on page 7

(m) *[2 marks]*

A queue based on a linked list uses the following code

```
class Node{
        String element;
        Node next;
        Node (String el, Node n)
        {
        element = el;
        next = n;
        }
}
Node front = null, rear = null;
```

What is the right code for String remove() operation? Such an operation removes and returns an element from the queue.

(i)

```
if (front == rear)
     throw new RuntimeException("Empty");
String temp = front.element;
front = front.next;
if (front == null)
     rear = null;
return temp;
```

(ii)

```
if (front == null)
     throw new RuntimeException("Empty");
String temp = front.element;
front = front.next;
if (front == null)
     rear = null;
return temp;
```

Question 1 (m) *(iii)* **is on page 8**

(iii)

```
if (rear== null)
     throw new RuntimeException("Empty");
String temp = rear.element;
rear = rear.next;
if (front == null)
     rear = null;
return temp;
```

(iv)

```
if (front == null)
     throw new RuntimeException("Empty");
String temp = front.element;
front = front.next;
if (front == null)
     front  = rear;
return temp;
```

(n) *[2 marks]*

Traversing a binary search tree in inorder

(i) yields the same sequence as a postorder traversal

(ii) yields a sequence in which the minimum element is at the approximate midpoint of the sequence

(iii) is more efficient than traversing the tree in preorder

(iv) will yield a sorted sequence

Question 1 (o) is on page 9

(o) *[2 marks]*

If ”14t8” is entered for input in the following code,

```
while (input != null)
{
        try
        {
        totalIncome += Double.parsedouble(input);
        months++;
        }
        catch(NumberFormatException e)
        {
        System.out.println("Non-numeric data encountered in the file: "
        + e.getMessage());
        }
        input = inputFile.readLine();
}
```

what does the program do?

(i) input will cause a NumberFormatError, the catch clause will be executed, then the program will resume with the statement following the while statement

(ii) input will be converted to a double and added to totalIncome, months will be incremented by 1, and the while statement will be repeated until a null is entered

(iii) input will cause a NumberFormatError, the catch clause will be executed, then the program will continue by asking for the next input value

(iv) input will cause a NumberFormatError, the catch clause will be executed, then the terminate

Question 1 (p) is on page 10

(p) *[2 marks]*

When a new item is added to an **AVL tree**

(i) each node has at least one predecessor and at most two successors

(ii) each node has at most one predecessor and at most one successor

(iii) each node has at most one predecessor and at most two successors

(iv) each node has at most one predecessor and exactly two successors

Question 2

Exceptions

(a) *[4 marks]*

Describe what an exception is.

(b) *[4 marks]*

How do you retrieve an error message from an exception?

(c) *[4 marks]*

What are the differences between a checked and an unchecked exception?

Question 3 is on page 11

Question 3

Generics *[3 marks]*

Consider the class

```
class Value <T extends Number>
{
    private T v;
    public Value(T v1)
    {
      v = v1;
    }
    public void output()
    {
      System.out.println(v);
    }
}
```

Explain in detail what the following statement does:

```
Value<Integer> nV1 = new Value<Integer>(37);
```

Question 4 is on page 12

Question 4

Recursion *[5 marks]*

What are overheads and how do they effect recursive algorithms?

Question 5

Collections

(a) *[3 marks]*

Why is it common practice to use an interface variable to reference a collection?

(b) *[3 marks]*

An array list has some advantages and disadvantages over a linked list.
What are the advantages and disadvantages?

(c) *[2 marks]*

The following code creates an ArrayList to hold String objects:

```
List<String> idNumbers = new ArrayList<String>();
```

Given that the following code returns a list iterator that is ready to be used with the ArrayList

```
ListIterator<String> it = nameList.listIterator();
```

Why is it **NOT** necessary to instantiate the list iterator yourself?

(d) *[2 marks]*

What is a hash code?

(e) *[2 marks]*

In hashing, what is a collision?

Question 6 is on page 13

Question 6

Array-Based Lists

(a) *[3 marks]*

Assume an array-based list implemented by a class that uses the fields

```
String [ ] list;
int nElements;
```

to represent the array of elements, and the number of elements currently stored.

Write the code for a method *int size*() that returns the current size of the list.

(b) *[6 marks]*

Briefly describe the operations specified by the `Iterator` interface.

(c) *[4 marks]*

What is the difference between the Iterator and Iterable interfaces?

Question 7

Linked Lists

[6 marks]

A linked list class uses a Node class with successor reference **next** and field **element** to store values. It uses a reference **first** to point to the first node of the list.
Correct the following code (rewrite it) so that it prints all elements stored in the list

```
Node p  = first;
while (p != null)
   System.out.println(p.next);
```

Question 8 is on page 14

Question 8

Stacks and Queues *[6 marks]*

Consider a class that uses the following variables to implement an array-based stack:

```
String [ ] s = new String[100];
int top = -1; //Note  top == -1 indicates stack is empty
```

Write the code for a method that implements a push operation. The method header for this operation is:

void push(String x)

(Note if the stack has no more room it should throw the appropriate exception)

Question 9

Trees *[11 marks]*

A binary tree can be represented by a non-linear dynamic data structure. Each node in the tree can have at most two offspring.

1. A binary search tree has the property that it is either empty or the root node's data value is larger than all of the data values contained in its left subtree and smaller than all the data values contained in its right subtree. Each subtree in a binary search tree must also be a binary search tree.

   Draw a diagram of the binary search tree that is generated by inserting the following characters in sequence:

   ```
   S Q J M W L U Y
   ```

Question 9 (a) 2. is on page 15

2. Show the order of visiting the nodes of the above binary tree using

   (a) *preorder traversal.*

   (b) *inorder traversal.*

   (c) *postorder traversal.*

`Please remember` – This examination question paper MUST BE HANDED IN. Failure to do so may result in the cancellation of all marks for this examination. Writing your name and number on the front will help us confirm that your paper has been returned.