UNIVERSITY OF NEW ENGLAND

NAME:____________________________________

STUDENT NUMBER:________________________

UNIT NAME: COMP132

PAPER TITLE: Computer Science II

PAPER NUMBER: First and Only

DATE: **Friday 12 November 2010** **TIME:** **9:15 AM TO 11:30 AM**

TIME ALLOWED: Two (2) hours and fifteen minutes

NUMBER OF PAGES IN PAPER: EIGHT (8)

NUMBER OF QUESTIONS ON PAPER: FIVE (5)

NUMBER OF QUESTIONS TO BE ANSWERED: FIVE (5)

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
- Answer the multiple choice questions by circling the letter that corresponds to the chosen answer on this examination paper
- Answer all other questions in the answer booklet provided
- Questions are NOT of equal value. Marks are given at the beginning of each question
- 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]*

How many times will the following method call itself, if 10 is passed as the argument?

```
public static void message(int n)
{
    if (n > 0)
    {
        System.out.println("Print this line.\n);
        message(n + 1);
    }
}
```

(i) 1

(ii) 9

(iii) 10

(iv) An infinite number of times

(b) *[2 marks]*

Which of the following problems cannot be programmed recursively?

(i) Towers of Hanoi

(ii) Greatest Common Denominator

(iii) Binary Search

(iv) All of these can be programmed recursively

Question 1 (c) is on page 3

(c) *[2 marks]*

The *QuickSort* algorithm works by

(i) repeatedly comparing adjacent items and swapping them so smaller values come before larger values

(ii) repeatedly locating the smallest value in the unsorted portion of the array and moving it toward the lower end of the array

(iii) repeatedly taking the first value in the unsorted portion of the array and placing it at its proper place in the part of the array that is already sorted

(iv) partitioning the unsorted portion of the array into two sublists and a pivot and recursively sorting the two sublists

(d) *[2 marks]*

The worst case complexity function is a good measure to use when:

(i) the load on the system is heaviest

(ii) we want a guarantee on the performance of an algorithm

(iii) the best case complexity would lead to incorrect results

(iv) we want to create an efficient algorithm

(e) *[2 marks]*

A class is *generic*

(i) if it is a subclass of the `Object` class

(ii) if it is a superclass of the `Object` class

(iii) if it has type parameters

(iv) if it has method parameters

(f) *[2 marks]*

When a generic class with an unconstrained type parameter is instantiated without specifying an actual type

(i) the type `Object` is used for the unspecified type

(ii) the compiler generates an error

(iii) the computer throws a `ClassCastException`

(iv) None of these

(g) *[2 marks]*

A collection whose elements are pairs of keys and values is called

(i) an enumerator

(ii) a map

(iii) a hash list

(iv) a paired collection

(h) *[2 marks]*

If different objects can have the same hash code, then

(i) some `HashSet` buckets may have more than one object

(ii) the `HashSet` add method will throw an `IllegalStateException`

(iii) the operation of searching a `HashSet` becomes more efficient

(iv) None of these

(i) *[2 marks]*

A new element is added to an `ArrayList` object at index k. Assuming the list has size s and does not have to be resized,

(i) the elements at current positions $0 \ldots k$ must be moved toward the beginning of the list

(ii) the elements at current positions $k \ldots s-1$ must be moved toward the end of the array

(iii) the elements at current positions $k \ldots s$ must be moved toward the end of the array

(iv) the element at position k is overwritten

(j) *[2 marks]*

A constructor for an array-based list takes an integer parameter, to be used as the capacity of the internal array of the list. Which exception should the constructor throw if its parameter is zero or negative?

(i) `IllegalArgumentException`

(ii) `IllegalStateException`

(iii) `RuntimeException`

(iv) `NullPointerException`

Question 1 (k) is on page 5

(k) *[2 marks]*

A linked list is represented by a reference to

(i) the first node in the list, unless the list is empty, in which case the reference is set to `null`

(ii) to the list representation object, which contains a boolean flag set to `false` when the list is empty

(iii) the superclass of the list

(iv) None of these

(l) *[2 marks]*

In a typical **circular doubly linked list**, a node has

(i) a field to store the element, and two references to keep track of two successor nodes, and a reference to keep track of the start of the list

(ii) a field to store the element, and two references to keep track of successor and predecessor nodes

(iii) a field to store the element, and two references to keep track of two predecessor nodes and a reference to keep track of the end of the list

(iv) either one of a field to store the element, and two references to keep track of two successor nodes, and a reference to keep track of the start of the list or a field to store the element, and two references to keep track of two predecessor nodes, and a reference to keep track of the end of the list

(m) *[2 marks]*

In an implementation of a stack, based on a singly-linked list, it is most efficient to add a new item so that

(i) the new item has the highest index of all items in the list

(ii) the new item has the lowest index of all items in the list

(iii) the new item is not duplicated by any other item already in the stack

(iv) the items in the stack stay sorted in ascending order

Question 1 (n) is on page 6

(n) *[2 marks]*

In a queue implementation that uses an array of fixed size

(i) the array must be made so large that the queue will never run out of space

(ii) it is necessary to use the array as a circular buffer

(iii) the array must be created from an `ArrayList` object

(iv) the queue must implement the `List` interface

(o) *[2 marks]*

A **binary tree** is a collection of nodes in which

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

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

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

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

(p) *[2 marks]*

A binary tree traversal method that visits the root first, and then recursively traverses the left and right subtrees is called

(i) top down traversal

(ii) priority order traversal

(iii) preorder traversal

(iv) postorder traversal

Question 2 *Exceptions*

(a) *[3 marks]*

Write a statement that throws an `IllegalArgumentException` with the error message “Argument cannot be negative”.

(b) *[6 marks]*

Explain the purpose of the *finally* clause. Illustrate your answer with an example application.

Question 3 is on page 7

Question 3 *Collections*

(a) *[4 marks]*

How does a map work?

(b) *[6 marks]*

How do you get an iterator for a list? Give an example.

(c) *[9 marks]*

Write an algorithm (or pseudo-code) for a method of a `HashSet` that adds a new object.

Question 4 *Lists and Stacks*

(a) *[6 marks]*

What are the three basic operations of an iterator?

(b) *[4 marks]*

Find an error in the following piece of code, describe it and provide a way to fix it:

```
// print all element in a list myList
Node ref = myList;
while (ref.next != null)
{
    System.out.print(ref.value + " ");
    ref = ref.next;
}
```

(c) *[8 marks]*

Suppose that you have two stacks but no queues. You have an application that needs to use a queue. Explain how to use two stacks to simulate a single queue.

Question 5 is on page 8

Question 5 *Trees*

(a) *[7 marks]*

How does an *AVL tree* differ from an ordinary binary search tree? Explain in 4–6 short sentences, list advantages and disadvantages of an AVL tree over a binary search tree. Illustrate your answer with an example of an AVL tree.

(b) *[7 marks]*

Give an algorithm (or a pseudo-code) for a method

```
int leafCount(Node bTree)
```

that computes and returns the number of leaves in a binary tree.

(c) *[8 marks]*

Define an order on strings where a string `left` is less than `right` if the length of `left` is smaller than the length of `right`. If `left` and `right` have the same length, then `left` is less than `right` if `left` comes before `right` in alphabetical order. Write a `Comparator` class that corresponds to this order.

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

.