



FACULTY OF SCIENCE

ACADEMY FOR INFORMATION TECHNOLOGY

MODULE CSC1B10
Introduction to data structures (C++)
CAMPUS APK

SSA EXAMINATION

DATE: 2014-12-04

SESSION 08:00 – 12:00

ASSESSOR(S)

DR DA COULTER

INTERNAL MODERATOR

DR DT VAN DER HAAR

DURATION 2 HOURS

MARKS 100

SURNAME, INITIALS (or ID NUMBER): _____

STUDENT NUMBER: _____

CONTACT NR: _____

NUMBER OF PAGES: 5 PAGES

REQUIREMENTS: NON-PROGRAMMABLE CALCULATORS ARE PERMITTED

Q1	
Q2	
Q3	
Q4	
Total	

QUESTION 1

Draw the following in your answer book					
1.1	Use UML to model the following scenario: A container has an iterator which must be capable of performing at least the following operations: increment and dereference. There are many different kinds of container for example vector, list and array.				(5)
Write the most correct option in your answer books					
1.2	Which of the following is a valid prototype destructor for the class Q1?				
	A ~Q1 ();	B @Q1 ();	C *Q1 ();	D \$Q1 ();	E ^Q1 ();
1.3	In order to declare a pure virtual function the following must be appended to a function prototype:				
	A = 0	B = NULL	C = nullptr	D virtual	E dynamic
1.4	It is safer to use protected inheritance than to use public inheritance.				
	True		False		無 ¹
1.5	Q1 operator ++ (); should return the ... state of the Q1 instance.				
	A stack allocated	B heap allocated	C original	D modified	E virtual
1.6	Q1 array[42] ; will call the Q1 class' ... constructor				
	A copy	B no-args	C base	D derived	E null
Write your answers to the following in your answer books					
1.7	What is the default visibility of the following: • class • struct				(2)
1.8	Describe two ways in which the separation of interface and implementation can be achieved in object oriented programming.				(2)
1.9	Define the following principles / techniques of object orientation • Polymorphism • Generics • Delegation				(3)
1.10	Name the programming technique associated with the following keywords: • try, catch, throw • virtual • operator				(3)

[20]

¹ The Sino-Japanese ideogram Wu/Mu in this case represents a question which is flawed.

QUESTION 2

Write your answers to the following in your answer books		
2.1	<p>Complete the following code:</p> <pre> template (a) void linkedListType<Type>::destroyList () { nodeType<Type> *temp; while (first (b) NULL) { temp = (c) first = (d) (e) temp; } last = NULL; count = 0; } </pre>	(5)
2.2	<p>Complete the following code:</p> <pre> const linkedListType<Type>& linkedListType<Type>::operator= (const (a) otherList) { if ((b) !=(c)otherList) { copyList((d)); } return (e) ; } </pre>	(5)
2.3	Write code to define a generic node type as used in Malik's doubly linked list types.	(5)
2.4	Describe how a range checked double indexing operator [] might be implemented which acts as both an accessor and mutator member function for two dimensional data (stored using dynamic arrays).	(5)
2.5	Due to their lack of support for random access not all sorting algorithms are suitable for linked lists. Provide pseudo-code for the bubble sort algorithm which can be used without random access.	(5)
2.6	Draw a series of images which depict the removal of the first node from a non-empty unordered singly linked list.	(5)

[30]**QUESTION 3**

In your answer books please write the necessary C++ code for the following statements, and answer the remaining questions. Unless otherwise indicated you may assume that the necessary header files are included. Most of the marks in this section are awarded for the file handling operations.		
3.1	<p>The text file <code>transactions.txt</code> is made up of lines in the following format:</p> <pre>ACCOUNTA ACCOUNTB AMOUNT EOL</pre>	(5)

	<p>ACCOUNTA and ACCOUNTB are textual representations of integers, AMOUNT is a representation of a double while EOL is the system's representation of the end of the line.</p> <p>Write code which reads in all of these lines and checks to make sure that the two accounts are not the same. For example if the text file contained...</p> <pre>12 14 2.0 12 12 5.5</pre> <p>... the output would be: OK ERROR</p> <p>Please show code for the opening / closing of the file.</p>	
3.2	Define a structure which stores the data from each line in the previous question. Show the line of code used to read one of these structures (called t1) from file called transactions.dat. You do not need to show the opening or closing of the file.	(5)
3.3	Assume that transactions.dat has had an unknown number of records written to it. Show how you would access a given record in the file by index while handling the potential for the file to enter into an error state.	(5)
3.4	Describe what the following IO modes mean <ul style="list-style-type: none"> • ios::in • ios::showpos • ios::ate • ios::app • ios::fixed 	(5)
3.5	<p>Consider the following function which searches through an unordered list.</p> <pre>template <class Type> bool unorderedLinkedList<Type>:: search(const Type& searchItem) const { nodeType<Type> *current; //pointer to traverse the list bool found = false; current = this->first; //set current to point to the first //node in the list while (current != NULL && !found) //search the list if (current->info == searchItem) //searchItem is found found = true; else current = current->link; //make current point to //the next node return found; } //end search</pre> <p>Analyse the above function asymptotically using Big-O notation. State any assumptions made.</p>	(10)

[30]

QUESTION 4

You need to access the action function which is located within the do.dll dynamically linked library. Unfortunately the header file and import library files for the DLL have gone missing so it is not possible to use implicit linking. In order to make use of the function you will need to write code which does the following:

The action function is a void function which takes a dynamically allocated array of characters as its input. The function is a C-function with no stated calling convention.

4.1	Includes the appropriate operating system library	(1)
4.2	Define an appropriate function pointer	(4)
4.3	Loads the library into an appropriate variable	(2)
4.5	Extracts the function from the library	(5)
4.5	Calls the function	(1)
4.6	Unloads the library	(2)
4.7	Describe how you could use exception handling with inheritance to robustly handle errors in the above process.	(5)

[20]