



Second Semester Examination
2018/2019 Academic Session

June 2019

CPT113/CPM213 – Programming Methodology & Data Structures
(Metodologi Pengaturcaraan & Struktur Data)

Duration : 2 hours
(Masa : 2 jam)

Please ensure that this examination paper contains **ELEVEN (11)** printed pages before you begin the examination.

*[Sila pastikan bahawa kertas peperiksaan ini mengandungi **SEBELAS (11)** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]*

Instructions: Answer all **FOUR (4)** questions.

Arahan: Jawab kesemua **EMPAT (4)** soalan.]

You may answer the questions either in English or in Bahasa Malaysia.

[Anda dibenarkan menjawab soalan sama ada dalam bahasa Inggeris atau bahasa Malaysia.]

In the event of any discrepancies, the English version shall be used.

[Sekiranya terdapat sebarang percanggahan pada soalan peperiksaan, versi bahasa Inggeris hendaklah diguna pakai.]

1.	Immigration Department of Malaysia, under Ministry of Home Affairs plan to develop an application for professional foreigner workers who are working in Malaysia to applying permanent resident status. There are few sets of data required in order to be able to process their application. Those data are foreigner information, job description and family information. Foreigner information consists of name, passport no. and highest qualification (PhD or Master). Job description includes type of job, position, year of service and yearly income. Family information consists of marital status and number of children. Design a solution using object oriented program approach, which demonstrates the implementation of basic object oriented, inheritance and composition . As a programmer, you should provide:	
	<i>Jabatan Imigresen Malaysia, dibawah Kementerian Dalam Negeri merancang untuk membangunkan sebuah applikasi bagi pekerja asing profesional untuk memohon status penduduk tetap. Terdapat beberapa set data yang diperlukan supaya dapat memproses aplikasi mereka. Data tersebut adalah maklumat orang asing, penerangan kerja dan maklumat keluarga. Maklumat orang asing mengandungi nama, no. passport dan kelayakan tertinggi (PhD or Master). Penerangan kerja termasuk jenis kerja, pangkat, tempoh perkhidmatan, dan pendapatan tahunan. Maklumat keluarga mengandungi status perkahwinan dan jumlah anak. Rekabentuk suatu penyelesaian dengan menggunakan asas berorientasikan objek, pewarisan dan komposisi. Sebagai juru atur cara, anda perlu menyediakan:</i>	
	(a)	Show UML diagram, which represents your class design. <i>Tunjukkan gambarajah UML yang mewakili rekabentuk kelas anda.</i> (5/200)
	(b)	Construct all required classes definition and the appropriate methods. <i>Hasilkan semua definisi kelas yang diperlukan dan kaedah-kaedah yang berkaitan.</i> (20/200)
	(c)	Construct main () function to do the following: <i>Hasilkan fungsi main () untuk melakukan yang berikut:</i>
	(i)	Declare an array of the object (size 200). <i>Isytihar suatu tatasusunan object (saiz 200).</i>

- 3-

		(ii)	<p>Read data from the text files (data.txt) and stores the required information into each array element of the object. Assume that each line in the text file represent one student's information. Name only consists of first name.</p> <p><i>Baca data dari fail teks (data.txt) dan simpan maklumat yang diperlukan ke dalam setiap elemen tatasusunan object. Anggapkan maklumat pada setiap baris dalam fail teks mewakili seorang pelajar. Nama hanya terdiri dari nama pertama sahaja.</i></p>
		(iii)	<p>Display foreigner information based on years of service. E.g. List all foreigner information which years of service more than 10 years.</p> <p><i>Papar maklumat orang asing berdasarkan masa perkhidmatan. Contoh. Senarai semua maklumat orang asing yang tempoh perkhidmatan lebih dari 10 tahun.</i></p> <p style="text-align: right;">(15/200)</p>

2.	(a)	<p>State the output of the following code.</p> <p><i>Tentukan output bagi kod berikut.</i></p> <pre> #include <iostream> using namespace std; int main () { int firstvalue = 10, secondvalue = 20; thirdvalue =30; fourthvalue= 40; int *p1, *p2, *p3, *p4; p1 = &firstvalue; p2 = &secondvalue; p3 = &thirdvalue; p4 = &fourthvalue; *p1 = 50; *p2 = *p1; p1 = p2; *p1 = 60; *p3 = 70; cout << "firstvalue is " << firstvalue << '\n'; cout << "secondvalue is " << secondvalue << '\n'; cout << "thirdvalue is " << thirdvalue << '\n'; cout << "value of *p2 is " << *p2 << '\n'; cout << "value of *p4 is " << *p4 << '\n'; return 0; } </pre> <p style="text-align: right;">(10/200)</p>
----	-----	--

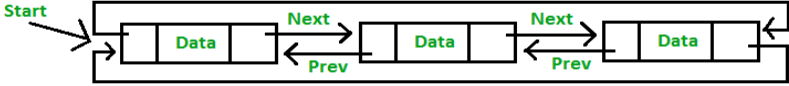
...4/-

	(b)	Construct C++ code for the following: <i>Hasilkan kod C++ untuk yang berikut:</i>
	(i)	Construct a dynamically allocated two dimensional array, having 4 rows and 5 columns of type int. <i>Cipta tatasusunan dinamik dua dimensi yang mempunyai 4 baris dan 5 lajur berjenis integer.</i> (4/200)
	(ii)	Produce the content for each element of the array by adding the row number and column number. For example, element (1,2) = 1+2 = 3. <i>Hasilkan kandungan setiap elemen tatasusunan ini dengan hasil jumlah nombor baris dan nombor lajur. Sebagai contoh, elemen (1,2) = 1+2 = 3.</i> (4/200)
	(iii)	Demonstrate how to destroy the dynamically allocated array. <i>Tunjukkan bagaimana untuk musnahkan tatasusunan dinamik yang telah diperuntukkan.</i> (2/200)
	(c)	Given 2 (two) linked lists, insert nodes of the second list into first list at alternate positions of the first list. Do not create additional nodes. For example, if first list is 5→7→17→13→11 and second list is 12→10→2→4→6 then the first list should become 5→12→7→10→17→2→13→4→11→6 and second list should become empty. The nodes of second list should only be inserted when there are positions available. Another example, if the first list is 1→2→3 and second list is 4→5→6→7→8 then first list should become 1→4→2→5→3→6 and second list is 7→8. Using the sample main program and sample output below, write in C++ code the following methods:

	<p><i>Diberi 2 (dua) senarai berpaut, selit nod-nod senarai kedua ke dalam senarai pertama pada kedudukan berselang senarai pertama. Jangan cipta nod tambahan.</i></p> <p><i>Sebagai contoh, jika senarai pertama ialah 5→7→17→13→11 dan senarai kedua ialah 12→10→2→4→6 maka senarai pertama menjadi 5→12→7→10→17→2→13→4→11→6 dan senarai kedua menjadi kosong. Nod senarai kedua hanya akan diselitkan apabila ada kekosongan.</i></p> <p><i>Contoh lain, jika senarai pertama ialah 1→2→3 dan senarai kedua ialah 4→5→6→7→8 maka senarai pertama menjadi 1→4→2→5→3→6 dan senarai kedua ialah 7→8.</i></p> <p><i>Menggunakan contoh atur cara utama dan contoh output di bawah, tulis kod C++ kaedah berikut:</i></p>
	<pre>// main program to test the following methods int main() { struct Node *p = NULL, *q = NULL; insert(&p, 3); insert(&p, 2); insert(&p, 1); cout<<"First Linked List:"<<endl; printList(p); // print the list insert(&q, 8); insert(&q, 7); insert(&q, 6); insert(&q, 5); insert(&q, 4); cout<<"Second Linked List:"<<endl; printList(q); // print the list merge(p, &q); cout<<"Modified First Linked List:"<<endl; printList(p); // print the list cout<<"Modified Second Linked List:"<<endl; printList(q); // print the list getchar(); return 0; }</pre> <p>Sample Output:</p> <pre>First Linked List: 1 2 3 Second Linked List: 4 5 6 7 8 Modified First Linked List: 1 4 2 5 3 6 Modified Second Linked List: 7 8</pre>

...6/-

		(i)	<p>Method <i>insert</i> to insert a node at the beginning of a linked list</p> <p><i>Kaedah insert yang menyelit nod pada permulaan senarai berpaut.</i></p> <p>(6/200)</p>
		(ii)	<p>Method <i>merge</i> that inserts nodes of linked list <i>q</i> into linked list <i>p</i> at alternate positions.</p> <p><i>Kaedah merge yang menyelit nod-nod senarai berpaut q ke dalam senarai berpaut p pada kedudukan berselang.</i></p> <p>(14/200)</p>
		(d)	<p>Given a Doubly Linked List data.</p> <p><i>Diberi data Senarai Berpaut Berganda.</i></p> <pre> class DoubleList{ // public methods declarations here ... private: struct DoubleNode{ ListItemType item; // data in the node DoubleNode* next; // points to next node DoubleNode* prev; // points to previous node }; DoubleNode* head; // points to first node DoubleNode* curr // points to a node }; </pre> <p>Assume the pointer <i>curr</i> points to any node in the doubly linked list. Write in C++ code to remove this node from the list. (What happen if <i>curr</i> is at the beginning or end of the list?</p> <p><i>Anggap penunding curr menunding kepada mana-mana nod dalam senarai berpaut berganda. Tulis kod C++ untuk hapuskan nod tersebut dari senarai. (Apa yang berlaku jika curr menunding pada permulaan atau penghujung senarai?)</i></p> <p>(10/200)</p>

	(e)	<p>Circular Doubly Linked List has properties of both doubly linked list and circular linked list in which two consecutive nodes are linked by previous and next pointer and the last node points to first node by next pointer and also the first node points to last node by previous pointer.</p> <p><i>Senarai Berganda Membulat mempunyai ciri-ciri kedua-dua senarai berganda dan senarai membulat, di mana nod yang berturutan dihubungkan dengan penunding sebelum dan selepas dan nod terakhir menunding kepada nod pertama oleh penunding selepas dan juga nod pertama menunjuk kepada nod terakhir oleh penunding sebelum.</i></p>  <p>Explain 1 (one) advantage, 1 (one) disadvantage and 1 (one) application of Circular Doubly Linked List.</p> <p><i>Terangkan 1 (satu) kebaikan, 1 (satu) keburukan dan 1 (satu) aplikasi Senarai Berganda Membulat.</i></p> <p style="text-align: right;">(10/200)</p>
3.		<p>Assuming there is a class called <i>qStack</i> which is a multiple inheritance class of queue and stack ADTs. Write a C++ code for the following:</p> <p><i>Andaikan ada satu kelas bernama qStack yang merupakan kelas pewarisan pelbagai daripada ADT baris-gilir dan tindanan. Tuliskan atur cara C++ untuk perkara berikut:</i></p>
	(a)	<p>(i) Declare an object of the derived class.</p> <p><i>Isytiharkan satu objek daripada kelas terbitan.</i></p> <p style="text-align: right;">(4/200)</p>
		<p>(ii) Using existing ADTs of both stack and queue, write a method to check whether the input is a palindrome. Palindrome is a word that reads the same backwards as forwards. E.g. madam.</p> <p><i>Menggunakan ADT yang wujud di dalam timbunan dan baris-gilir, tuliskan satu kaedah untuk memeriksa sama ada input yang diterima adalah satu palindrom. Palindrom merupakan satu perkataan yang sama apabila dieja dari kiri atau kanan. Contohnya: madam.</i></p> <p style="text-align: right;">(13/200)</p>

		<p>(iii) Write a method that check the content of a queue and if it is not a prime number, it is removed from the queue. You are allowed to use two queue objects.</p> <p><i>Tuliskan satu kaedah yang memeriksa isi kandungan satu baris-gilir dan jika ia bukan nombor prima, nombor tersebut dikeluarkan daripada baris gilir tersebut. Anda dibenarkan menggunakan dua objek baris-gilir</i></p> <p>(13/200)</p>
	(b)	<p>Given queue and stack ADTs, briefly and accurately explain what are the following program is trying to do:</p> <p><i>Diberi ADT baris-gilir dan tindanan, jelaskan dengan ringkas dan tepat apakah yang sedang dilakukan oleh aturcara yang berikut:</i></p>
	(i)	<pre> linkedStack stack; int first = 0, second = 1, next; for (int i = 0 ; i < 10 ; i++) { if (i <= 1) next = i; else { next = first + second; first = second; second = next; } stack.push(next); } while (!stack.isEmptyStack()) { cout << stack.top() << endl; stack.pop(); } </pre> <p>(10/200)</p>
	(ii)	<pre> while (!stack.isEmptyStack()) { queue.addQueue(stack.top()); stack.pop(); } while (!queue.isEmptyQueue()) { cout << queue.front() << " "; queue.deleteQueue(); } </pre> <p>(10/200)</p>

		(iii)	<pre> for (int i = 1 ; i <= 10 ; i++) { queue.addQueue(i*i); } while (!queue.isEmptyQueue()) { if (queue.front() % 2 == 0) { stack.push(queue.front()); queue.deleteQueue(); } else { cout << queue.front() << " "; queue.deleteQueue(); } } </pre> <p style="text-align: right;">(10/200)</p>
4.	(a)		<p>The following function $C(n,r)$ denotes the number of ways (r) can be chosen from a set of n items, where r and n are nonnegative integers and $r \leq n$.</p> <p><i>Fungsi $C(n,r)$ berikut menandakan bilangan cara (r) boleh dipilih dari satu set item n, di mana r dan n adalah nombor integer bukan negatif dan $r \leq n$.</i></p> $c(n,r) = \begin{cases} 1 & \text{if } r = 0 \text{ or } n = r \\ n & \text{if } r = 1 \\ c(n-1, r-1) + c(n-1, r) & \text{otherwise} \end{cases}$
		(i)	<p>Construct a C++ recursive algorithm to determine $C(n,r)$.</p> <p><i>Hasilkan algoritma rekursi C++ untuk tentukan $C(n,r)$.</i></p> <p style="text-align: right;">(10/200)</p>
		(ii)	<p>Analyze the output for the recursive call $C(5,3)$. Show your working.</p> <p><i>Analisa output bagi panggilan rekursi $C(5,3)$. Tunjuk jalan kerja anda.</i></p> <p style="text-align: right;">(10/200)</p>

(b)	<p>The following lists the nodes in a binary tree in two different orders:</p> <p><i>Berikut adalah senarai nod pepohon penduaan dalam dua tertib yang berbeza:</i></p> <p>Preorder: b e s t o f l u c k Inorder : t s e f o l b u k c</p> <p><i>Tertib awalan: b e s t o f l u c k</i> <i>Tertib sisipan: t s e f o l b u k c</i></p>
(i)	<p>Form the original binary search tree.</p> <p><i>Bentukkan pepohon gelintaran penduaan yang asal.</i></p> <p style="text-align: right;">(10/200)</p>
(ii)	<p>Give the postorder traversal.</p> <p><i>Beri susuran tertib akhiran.</i></p> <p style="text-align: right;">(6/200)</p>
(c)	<p>Given the binary tree, answer the following questions:</p> <p><i>Diberi pepohon penduaan, jawab soalan-soalan berikut:</i></p>

- 11 -

		(i)	Identify the height of the tree with root 72. <i>Tentukan ketinggian pepohon dengan akar 72.</i>
		(ii)	Identify the level of the node 35. <i>Tentukan aras nod 35.</i> (4/200)