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 THIRTEEN (13) printed pages before you begin the examination.

[Sila pastikan bahawa kertas peperiksaan ini mengandungi TIGA BELAS (13) 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 the Ministry of Home Affairs plan to develop an application for professional foreign workers who are working in Malaysia to apply for permanent resident status. A few sets of data are required in order to process their applications. The data are applicant's information, job description and family information. Applicant's information consists of name, passport number and highest qualification (PhD or Master). Job description includes type of job, position, years of service and yearly income. Family information consists of marital status and the number of children. Design a solution using object oriented programming approach, which demonstrates the implementation of **basic object oriented**, **inheritance** and **composition**. As a programmer, you should do the followings:

   Jabatan Imigresen Malaysia, di bawah Kementerian Dalam Negeri merancang untuk membangunkan sebuah aplikasi bagi pekerja asing profesional untuk memohon status penduduk tetap. Beberapa set data diperlukan untuk memproses aplikasi mereka. Data tersebut adalah maklumat pemohon, penerangan kerja dan maklumat keluarga. Maklumat pemohon mengandungi nama, nombor pasport dan kelayakan tertinggi (PhD atau Master). Penerangan kerja termasuk jenis kerja, pangkat, tempoh perkhidmatan, dan pendapatan tahunan. Maklumat keluarga terdiri daripada status perkahwinan dan jumlah anak. Reka bentukkan suatu penyelesaian dengan menggunakan pendekatan pengaturcaraan berorentasikan objek, pewarisan dan komposisi. Sebagai pengatur cara, anda perlu lakukan seperti berikut:

   (a) Show UML diagram that represents your class design.

   Tunjukkan gambar rajah UML yang mewakili reka bentuk kelas anda.

   (5/200)

   (b) Construct all required classes definition and the appropriate methods.

   Bina semua definisi kelas yang diperlukan dan kaedah-kaedah yang bersesuaian.

   (20/200)

   (c) Construct `main()` function to do the following:

   Bina fungsi `main()` untuk melakukan yang berikut:

(i) Declare an array of the object of size 200.

Isytihar satu tatasusunan objek bersaiz 200.

(ii) Read data from the text files (data.txt) and store the required information into each array element of the object. Assume that each line in the text file represents one applicant's information. Name of the applicant consists of first name only.

Baca data dari fail teks (data.txt) dan simpan maklumat yang diperlukan ke dalam setiap elemen tatasusunan objek. Anggapkan maklumat pada setiap baris dalam fail teks mewakili seorang pemohon. Nama pemohon hanya terdiri dari nama pertama sahaja.

(iii) Display applicant information based on years of service. Example: List all applicant information which years of service more than 10 years.

Papar maklumat pemohon berdasarkan jumlah tahun perkhidmatan. Contoh: Senaraikan semua maklumat pemohon yang tempoh perkhidmatan lebih daripada 10 tahun.

(15/200)

2. (a) State the output of the following code.

Nyatakan output bagi kod berikut.

```
#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;
}
```

(10/200)

(b) Construct C++ code for the following:

Bina kod C++ untuk yang berikut:

(i) A dynamically allocated two-dimensional array, having 4 rows and 5 columns of type int.

Tatasusunan dua dimensi yang diperuntukkan secara dinamik yang mempunyai 4 baris dan 5 lajur berjenis int.

(ii) Produce the content for each element of the array by adding the row number and column number. For example, if row number is **one (1)** and column number is **two (2)**, then the value of that specific element is 1+2 = 3.

*Hasilkan kandungan setiap elemen tatasusunan ini dengan menambah nombor baris dan nombor lajur. Sebagai contoh, jika nombor baris ialah **satu (1)** dan nombor lajur ialah **dua (2)**, maka nilai elemen tersebut ialah 1+2 = 3.*

(iii) Demonstrate how to destroy the dynamically allocated array.

Tunjukkan bagaimana untuk memusnahkan tatasusunan yang telah diperuntukkan secara dinamik.

(10/200)

(c) Given **two (2)** linked lists, you are required to insert nodes of the second list into first list at alternate positions of the first list. Do not create additional nodes.

If first list is 5→7→17→13 and second list is 12→10→2→4 then the first list should become 5→12→7→10→17→2→13→4 and second list should become empty.

If first list is 1→2 and second list is 3→4→5→6 then first list should become 1→3→2→4 and second list is 5→6.

Given below, the sample main program and the sample output:

*Diberi **dua (2)** senarai berpaut, anda diperlukan untuk menyelit nod-nod senarai kedua ke dalam senarai pertama secara berselang seli dalam senarai pertama. Jangan cipta nod tambahan.*

Jika senarai pertama ialah 5→7→17→13 dan senarai kedua ialah 12→10→2→4 maka senarai pertama menjadi 5→12→7→10→17→2→13→4 dan senarai kedua menjadi kosong.

Jika senarai pertama ialah 1→2 dan senarai kedua ialah 3→4→5→6 maka senarai pertama menjadi 1→3→2→4 dan senarai kedua ialah 5→6.

Diberi contoh atur cara utama dan contoh output di bawah:

```
struct nodeList
{
   int data;
   nodeList *next;
};
int main()
{
    struct nodeList *p = NULL, *q = NULL;
    insert(&p, 2);
    insert(&p, 1);
    cout<<"First Linked List:"<<endl;
    printList(p); // print the list

    insert(&q, 6);
    insert(&q, 5);
    insert(&q, 4);
    insert(&q, 3);
    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
    return 0;
}
```

Sample Output:

Contoh Output:

```
First Linked List:
1 2
Second Linked List:
3 4 5 6
Modified First Linked List:
1 3 2 4
Modified Second Linked List:
5 6
```

Write in C++:

Tulis dalam C++:

(i) A method `insert` to insert a node with data at the beginning of a linked list.

Kaedah `insert` yang menyelit nod bersama data pada permulaan senarai berpaut.

(ii) A method `merge` that merges nodes of linked list q into linked list p at alternate positions as described above.

Kaedah `merge` yang mencantum nod-nod senarai berpaut q ke dalam senarai berpaut p pada kedudukan berselang seli seperti yang diterangkan di atas.

(20/200)

(d) Given a segment of doubly-linked list.

Diberi segmen senarai berpaut berganda.

```
struct nodeDList
{
   int data; // data in the node
   nodeDList *next; // points to next node
   nodeDList *prev; // points to previous node
};
class doublyLinkedList
{
// public methods declarations here...
   public:
      void reversePrint()const // output data in reverse order
   protected:
      nodeDList *first; // points to first node
      nodeDList *last; // points to last node
};
```

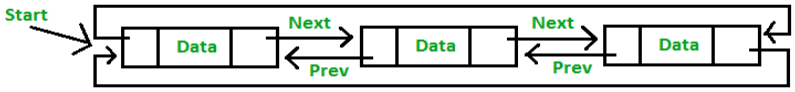
Write in C++ the method `reversePrint` to print the data contained in each node in reverse order, starting from the last node of the list.

Tulis dalam C++ kaedah `reversePrint` *untuk mencetak data yang terkandung dalam setiap nod tersebut secara terbalik, bermula dari nod yang terakhir senarai berkenaan.*

(10/200)

(e) A circular doubly-linked list has properties of both doubly-linked list and circular-linked list in which two consecutive nodes are linked by the previous and the next pointers, and the last node points to first node by the next pointer and also the first node points to last node by the previous pointer.

Senarai berganda membulat mempunyai ciri kedua-dua senarai berganda dan senarai membulat, iaitu nod yang berturutan dihubungkan dengan penuding sebelum dan selepas, dan nod terakhir menuding kepada nod pertama oleh penuding selepas dan juga nod pertama menunjuk kepada nod terakhir oleh penuding sebelum.



Explain **one (1)** advantage, **one (1)** disadvantage and **one (1)** application of circular doubly-linked list.

*Terangkan **satu (1)** kebaikan, **satu (1)** keburukan dan **satu (1)** aplikasi senarai berganda membulat.*

(10/200)

3. Assuming there is a class called *qStack*, which is a multiple inheritance class of queue and stack ADTs. Write a C++ code for the following:

Andaikan ada satu kelas bernama qStack yang merupakan kelas pewarisan pelbagai daripada ADT baris gilir dan tindanan. Tulis atur cara C++ untuk perkara berikut:

(a) (i) Declare an object of the derived class.

Isytiharkan satu objek daripada kelas terbitan berkenaan.

(ii) Using existing ADTs of both stack **and** queue, write a method to check whether the input is a palindrome. The method can be either a member or non-member of *qStack* class. Palindrome is a word that reads the same backwards as forwards. For example: madam.

*Menggunakan ADT yang wujud dalam tindanan **dan** baris gilir, tulis satu kaedah untuk memeriksa sama ada input yang diterima adalah satu palindrom. Kaedah tersebut boleh menjadi ahli atau bukan ahli kelas qStack. Palindrom merupakan satu perkataan yang sama apabila dieja dari kiri atau kanan. Contohnya: madam.*

(iii) Write C++ code that checks 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 (2)** queue objects.

*Tulis kod C++ yang memeriksa isi kandungan satu baris gilir dan jika nombor berkenaan bukan nombor perdana, nombor berkenaan dikeluarkan daripada baris gilir tersebut. Anda dibenarkan menggunakan **dua (2)** objek baris gilir.*

(30/200)

(b) Given queue and stack ADTs, briefly and accurately explain what the following program is trying to do:

Diberi ADT baris gilir dan tindanan, jelaskan dengan ringkas dan tepat apakah yang sedang dilakukan oleh atur cara yang berikut:

(i)
```
linkedStackType 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();
   }
```

(ii)
```
while (!stack.isEmptyStack())
   {
      queue.addQueue(stack.top());
      stack.pop();
   }

   while (!queue.isEmptyQueue())
   {
      cout << queue.front() << " ";
      queue.deleteQueue();
```

```
}
```


(iii)
```
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();
      }
   }
```

(30/200)

4. (a) 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 <= n`.

Fungsi `C(n,r)` berikut menandakan bilangan cara `(r)` boleh dipilih dari satu set item n, yang mana `r` dan n adalah nombor integer bukan negatif dan `r <= n`.

$$c(n,r)=\begin{cases} 1 & if\ r=0\ or\ n=r \\ n & if\ r=1 \\ c(n-1,r-1)+c(n-1,r) & otherwise \end{cases}$$

(i) Construct a C++ recursive function to determine `C(n,r)`.

Hasilkan fungsi rekursi C++ untuk tentukan `C(n,r)`.

(ii) Analyze the output for the recursive function call `C(5,3)`. Show your working.

Analisiskan output bagi panggilan fungsi rekursi `C(5,3)`. Tunjuk jalan kerja anda.

(20/200)

...12/-

...13/-

(b) The following lists the nodes in a binary tree in two different orders:

Berikut adalah senarai nod pepohon perduaan dalam dua tertib yang berbeza:

Preorder: b e s t o f l u c k
Inorder : t s e f o l b u k c

Tertib awalan: b e s t o f l u c k
Tertib sisipan: t s e f o l b u k c

Based on the given order:

Berdasarkan tertib yang diberikan:

(i) Form the original binary search tree.

Bentukkan pepohon gelintaran perduaan yang asal.

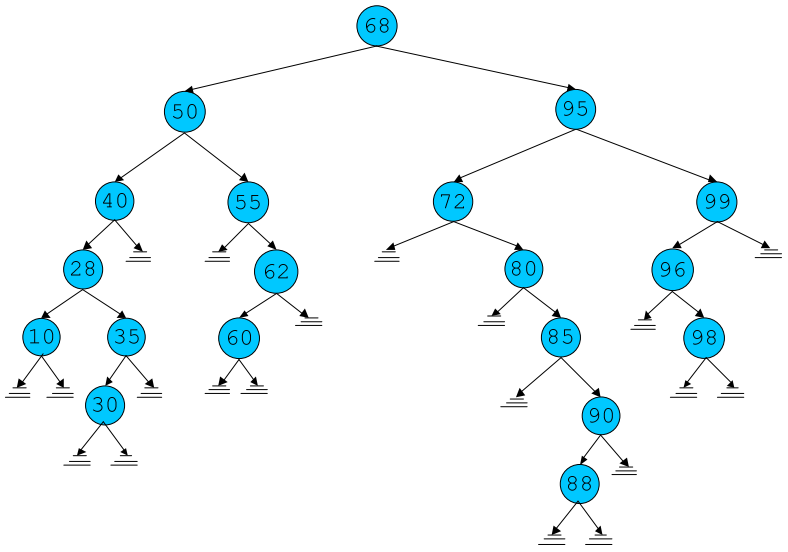
(ii) Give the postorder traversal.

Beri penyusuran tertib akhiran.

(16/200)

(c) Given the binary tree below, answer the following questions:

Diberi pepohon perduaan dibawah, jawab soalan-soalan berikut:



(i) Identify the height of the tree with root 72.

Kenal pasti ketinggian pepohon dengan akar 72.

(ii) Identify the level of the node 35.

Kenal pasti aras nod 35.

(4/200)

- oooOooo -