

---

UNIVERSITI SAINS MALAYSIA

Second Semester Examination  
2014/2015 Academic Session

June 2015

**CPT113/CPM213 – Programming Methodology & Data Structures**  
*[Metodologi Pengaturcaraan & Struktur Data]*

Duration : 2 hours  
*[Masa : 2 jam]*

---

**INSTRUCTIONS TO CANDIDATE:**

***[ARAHAN KEPADA CALON:]***

- Please ensure that this examination paper contains **THREE** questions in **FIFTEEN** printed pages before you begin the examination.

*[Sila pastikan bahawa kertas peperiksaan ini mengandungi **TIGA** soalan di dalam **LIMA BELAS** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]*

- Answer **ALL** questions.

*[Jawab **SEMUA** 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. (a) Given the following class definition:

```
class contactInfor {
    string name;
    int phone;
public :
    void setContact(string CName,int phone){
        name=CName;
        phone=phone;}

    void printContact(){
        cout <<"Contact person infor: " << endl;
        cout << "\t" << "Name : " << name << endl;
        cout << "\t" << "Phone: " << phone << endl; }
};
```

Pink Rose Holiday is a new tour company that provides tour services for group tours. The company would like to develop the system which automatically computes the package tour price for the group and also store a contact person information. Therefore, you are required to write a program based on the following requirements:

- (i) Define the class named `travelPackage` which consist of the following class members:
- Default constructor and destructor.
  - Function prototype `setPackage` which receives values such as a package price of a person, number of people in a group and GST tax rate.
  - Function prototype `displayPackage` to display a package price of a person, number of people in the group and GST tax rate.
  - Function prototype `getPrice`. This function returns the package price of a person.
  - Function prototype `getPerson`. This function will return the number of people in a group.
  - Function prototype `getTax`. This function will return the tax.
  - Variables declaration: `price`, `no_person` and `GST` as private members.
- (ii) Write the function definitions `setPackage` and `displayPackage` in the class `travelPackage`.

(iii) Define the class named `packagePrice`. The class `packagePrice` inherits as a public from the class `travelPackage`. The class `packagePrice` consists of the following members:

- Default constructor and destructor.
- Function prototype `computerPrice` that returns the total price of the package of the group by considering the number of people in a group and GST tax rate.  
[Tax is calculated on the top of package price of the group].
- Function prototype `setInfor` which will receive the values of contact person name and phone number. This function will store the received values into appropriate variable in the class `contactInfor`.
- function prototype `displayContact` that displays name and phone number of the contact person.
- Variables declaration: `total` to store total price of the package and `person` as an object typed of the class `contactInfor` to store contact information as private members.

(iv) Write the function definitions `computerPrice` and `contactInfor` in the class `packagePrice`.

(v) Complete the following `main()` function:

```
main() {
    (a) Declare the object named pkg typed packagePrice
        string firstN; int phone;
        pkg.setPackage(1800,2,0.06);
        cout <<"Enter Contact Person'name: ";
        cin >> firstN;
        cout <<"Enter Phone: ";
        cin >> phone;
    (b) Call function to store the entered information
    (c) Call function to display contact person information
        pkg.displayPackage();
        cout << "Total Price to be paid: " ;
    (d) Display total price of the package
}
```

(26/100)

(b) Answers the following questions:

- (i) Explain the purpose of constructors and destructors.
- (ii) Explain the usage of `friend` function.
- (iii) Explain how the private members of a class can be accessed from the derived class?

(9/100)

2. (a) Answer the following questions:

(i) Explain what is wrong with the following code segment?

```

int *p; //Line 1
int *q; //Line 2

p = new int [5]; //Line 3
*p = 2; //Line 4

for(int i = 1; i < 5; i++) //Line 5
    p[i] = p[i-1] + i; //Line 6

q = p; //Line 6

delete [] p; //Line 7

for(int j = 0; j < 5; j++) //Line 8
    cout<<q[j]<<" "; //Line 9

cout<<endl; //Line 10

```

(ii) What is the output of the following code?

```

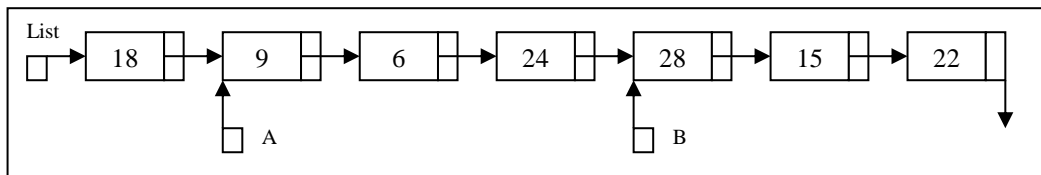
int *p;
int *q;
p = new int;
*p = 43;
q=p;
*q = 52;
p = new int;
*p = 78;
q = new int;
*q = *p;
cout<< *p << " " << *q << endl;

```

(iii) What is a copy constructor and state **two (2)** situations where copy constructors are executed?

(10/100)

- (b) Consider the linked list shown below: (Assume that `List`, `A` and `B` are external pointers of node. Every node contains integer value in `info` and a pointer `link`).



What is the output of the following C++ statements?

- (i) `cout<<List->info;`
- (ii) `cout<<A->info;`
- (iii) `cout<<B->link->info;`
- (iv) `cout<<List->link->link->link->info;`

Write C++ statement to do the following, each statement is executed separately.

- (v) Make `A` points to the node containing value 24.
- (vi) Make `List` point to an empty list.
- (vii) Make `B` point to the last node in the list.
- (viii) Change the `info` value of third node in the linked list to 3.

(8/100)

- (c) Given a circular linked list of strings that are sorted into **ascending** order as being illustrated in Figure 1. The external pointer `list` points to the last node, which contains the largest string. Write a function that reorganises the list so that its data are sorted into **descending** order as being illustrated in Figure 2. Do not allocate or deallocate nodes.

**Note:** You do not have to provide a complete method or class. You may declare more variables as needed.

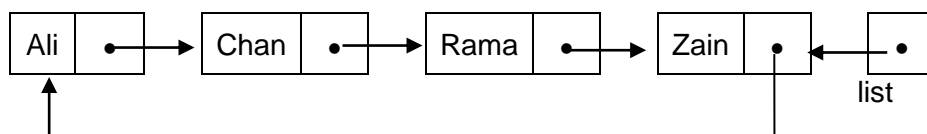


Figure 1

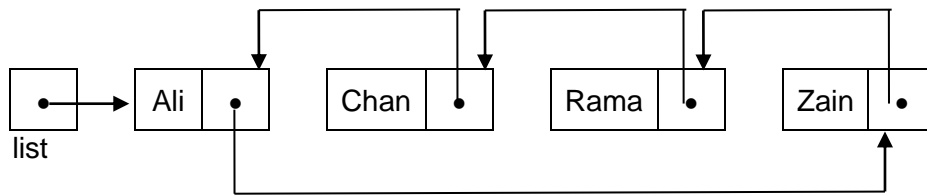


Figure 2

(10/100)

(d) Consider the following recursive function:

```
int myFunc(int x)
{
  if (x == 0)
    return 2;
  else if (x == 1)
    return 3;
  else
    return (myFunc (x - 1) + myFunc (x - 2));
}
```

Answer the following questions:

- (i) State the base and general cases.
- (ii) What is the output of statement `cout << myFunc(5) << endl;`? Show your working.

(7/100)

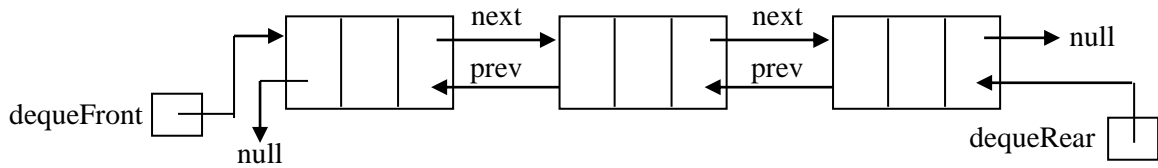
3. (a) Palindrome is a situation where a text reads the same if it is read forward or backward. For example:

A Toyota: Race fast safe car: a Toyota

Write in C++ the main function that reads a line of text and place each letter onto both a stack and a queue, then verify whether the text is a palindrome. Use existing stack and queue functions.

(8/100)

- (b) Deque is a double-ended queue, which allows addition and deletion at both ends. Assume it to be implemented as double linked list as described below:



```

template <class Type>
struct nodeType
{
    Type info;
    nodeType<Type> *next;
    nodeType<Type> *prev;
};

template<class Type>
class linkedDequeType
{
public:
    //All functions for Deque including ...
    void addDequeFront(const Type& newItem);
    //Add newItem in the front node
    void addDequeRear(const Type& newItem);
    //Add newItem in the back node
    void deleteDequeFront ( );
    //Delete an item from the front node
    void deleteDequeRear( );
    //Delete an item from the back node
    ~linkedDequeType();
    //destructor

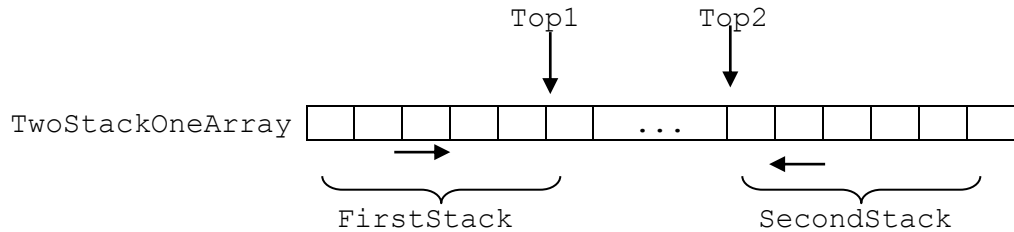
private:
    nodeType<Type> *dequeFront;
    //pointer to the front node
    nodeType<Type> *dequeRear;
    //pointer to the back node
};

```

Write in C++ the function definition for `deleteDequeRear ( ) ;`.

(8/100)

- (c) The storage structure of using two stacks `FirstStack` and `SecondStack` whose elements are of the same type would consist of a single array `TwoStackOneArray` and two top pointers; `Top1` and `Top2` which let the two stacks grow towards each other; as described below:



- (i) Write the function definition for `push` using dynamic array. The function should receive as parameter; which stack to be operated upon.
- (ii) State the condition when `push` function fails.

(7/100)

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

```
preorder: 49 28 18 11 19 40 32 44 83 71 69 72 97 92 99
inorder:  11 18 19 28 32 40 44 49 69 71 72 83 92 97 99
```

- (i) Draw the original binary search tree.
- (ii) List the postorder traversal.
- (iii) Which traversals always visit:
  - the left-most node first?
  - the right-most node last?

(7/100)



1. (a) Diberi definisi-definisi kelas berikut:

```
class contactInfor {
    string name;
    int phone;
public :
    void setContact(string CName,int phone){
        name=CName;
        phone=phone;}

    void printContact(){
        cout <<"Contact person infor: " << endl;
        cout << "\t" << "Name : " << name << endl;
        cout << "\t" << "Phone: " << phone << endl; }
};
```

Pink Rose Holiday adalah syarikat pelancongan baharu yang menawarkan perkhidmatan kepada kumpulan pelancong. Syarikat ini hendak menghasilkan sebuah sistem yang mengira secara automatik harga pakej pelancongan kumpulan pelancong dan juga menyimpan maklumat orang yang dihubungi. Maka oleh yang demikian, anda dikehendaki menulis program berdasarkan keperluan yang berikut:

- (i) Definiskan kelas `travelPackage` yang mengandungi ahli-ahli kelas yang berikut:
- Pembina lalai dan pemusnah lalai.
  - Fungsi prototaip `setPackage` yang menerima nilai seperti harga pakej untuk seorang, bilangan orang dalam kumpulan dan kadar cukai. GST.
  - Fungsi prototaip `displayPackage` untuk memaparkan harga pakej untuk seorang, bilangan orang dalam kumpulan dan kadar cukai GST.
  - Fungsi prototaip `getPrice`. Fungsi ini memulangkan harga pakej untuk seorang.
  - Fungsi prototaip `getPerson`. Fungsi ini akan memulangkan bilangan orang dalam kumpulan.
  - Fungsi prototaip `getTax`. Fungsi ini akan memulangkan cukai.
  - Pengisytiharan pemboleh ubah: `price`, `no_person` dan `GST` sebagai ahli persendirian.
- (ii) Tulis implementasi fungsi `setPackage` dan `displayPackage` dalam kelas `travelPackage`.

- (iii) Definiskan kelas yang bernama `packagePrice`. Kelas `packagePrice` mewarisi secara awam dari kelas `travelPackage`. Kelas `packagePrice` mengandungi ahli-ahli kelas yang berikut:
- Pembina lalai dan pemusnah lalai.
  - Fungsi prototaip `computerPrice` yang memulangkan jumlah harga pakej dalam kumpulan dengan mengambil kira bilangan orang dalam kumpulan dan kadar cukai GST.  
[Cukai dikira bersama-sama dengan harga pakej kumpulan].
  - Fungsi prototaip `setInfor` yang akan menerima nilai nama orang yang dihubungi dan nombor telefon. Fungsi ini akan menyimpan nilai yang diterima ke dalam pemboleh ubah yang sesuai dalam kelas `contactInfor`.
  - Fungsi prototaip `displayContact` yang memaparkan nama dan nombor telefon orang yang dihubungi.
  - Pengisytiharan pemboleh ubah: `total` untuk menyimpan jumlah harga pakej dan `person` sebagai jenis objek kelas `contactInfor` untuk menyimpan maklumat kontak sebagai ahli persendirian.
- (iv) Tulis definisi fungsi `computerPrice` dan `contactInfor` dalam kelas `packagePrice`.
- (v) Lengkapkan fungsi `main( )` yang berikut:

```
main() {
    (a) Isytiharkan objek yang bernama pkg jenis packagePrice
        string firstN; int phone;
        pkg.setPackage(1800,2,0.06);
        cout <<"Enter Contact Person'name: ";
        cin >> firstN;
        cout <<"Enter Phone: ";
        cin >> phone;
    (b) Memanggil fungsi untuk menyimpan maklumat yang dimasukkan
    (c) Memanggil fungsi untuk memaparkan maklumat orang yang
        dihubungi
        pkg.displayPackage();
        cout << "Total Price to be paid: " ;
    (d) Memaparkan jumlah harga pakej
}
```

(26/100)

- (b) Jawab soalan yang berikut:

- (i) Jelaskan tujuan pembina lalai dan pemusnah lalai.
- (ii) Jelaskan penggunaan fungsi kawan.
- (iii) Jelaskan bagaimana ahli persendirian dalam sebuah kelas boleh dicapai dari kelas terbitan?

(9/100)

2. (a) Jawab soal-soal ringkas yang berikut:

(i) Apakah kesalahan yang terdapat di dalam pecahan kod berikut?

```

int *p; //Line 1
int *q; //Line 2

p = new int [5]; //Line 3
*p = 2; //Line 4

for(int i = 1; i < 5; i++) //Line 5
    p[i] = p[i-1] + i; //Line 6

q = p; //Line 6

delete [] p; //Line 7

for(int j = 0; j < 5; j++) //Line 8
    cout<<q[j]<<" "; //Line 9

cout<<endl; //Line 10

```

(ii) Apakah output bagi kod berikut?

```

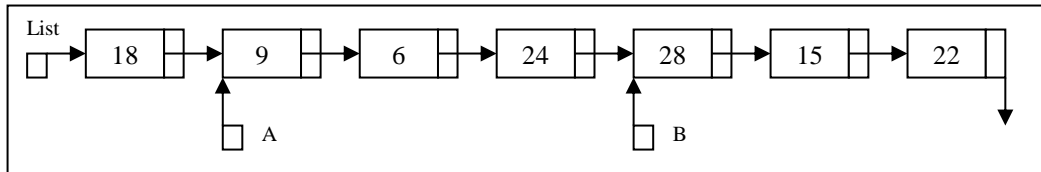
int *p;
int *q;
p = new int;
*p = 43;
q=p;
*q = 52;
p = new int;
*p = 78;
q = new int;
*q = *p;
cout<< *p << " " << *q << endl;

```

(iii) Apakah fungsi pembina salinan dan namakan **dua** (2) situasi di mana pembina salinan akan dilaksanakan?

(10/100)

- (b) Diberi senarai berpaut seperti di bawah. (Anggap `List`, `A` dan `B` adalah penuding luaran kepada nod. Setiap nod mengandungi nilai integer dalam `info` dan penuding `link`).



Apakah yang dicetak oleh kenyataan C++ berikut?

- (i) `cout<<List->info;`
- (ii) `cout<<A->info;`
- (iii) `cout<<B->link->info;`
- (iv) `cout<<List->link->link->link->info;`

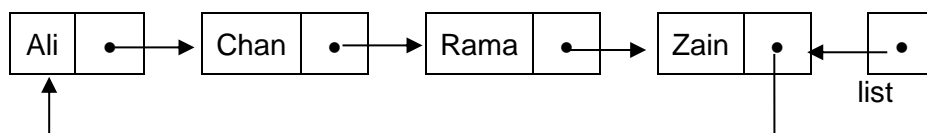
Tulis kenyataan C++ untuk melakukan arahan berikut, setiap kenyataan dilaksanakan secara berasingan.

- (v) Penuding `A` menuding kepada nod bernilai 24.
- (vi) Penuding `List` menuding kepada senarai kosong.
- (vii) Penuding `B` menuding kepada nod terakhir dalam senarai.
- (viii) Tukar nilai dalam nod `info` yang ketiga kepada nilai 3.

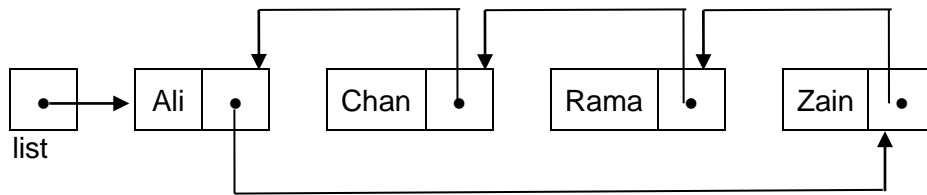
(8/100)

- (c) Diberi senarai rentetan berpaut membulat yang tersusun secara **menaik** seperti gambaran Rajah 1. Penuding luaran `List` menuding kepada nod terakhir, yang mengandungi rentetan terbesar. Tulis fungsi yang mengubah suai senarai supaya data disusun secara **menurun** seperti gambaran dalam Rajah 2. Jangan peruntuk atau hapus nod-nod.

**Nota:** Anda tidak perlu menyediakan satu kaedah atau kelas yang lengkap. Anda boleh mengisytiharkan lebih pemboleh ubah yang diperlukan.



Rajah 1

Rajah 2

(10/100)

(d) Pertimbangkan fungsi rekursif yang berikut:

```
int myFunc(int x)
{
  if (x == 0)
    return 2;
  else if (x == 1)
    return 3;
  else
    return (myFunc (x - 1) + myFunc (x - 2));
}
```

Jawab soalan-soalan berikut:

- (i) Nyatakan kes-kes asas dan umum.
- (ii) Apakah output bagi kenyataan `cout << myFunc(5) << endl;?` Tunjukkan jalan kerja anda.

(7/100)

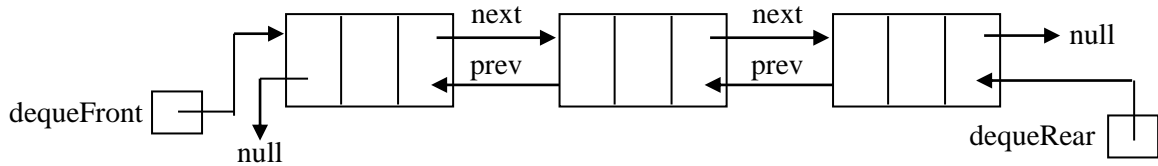
3. (a) Palindrom adalah situasi di mana teks yang bermaksud sama jika ia dibaca dari depan atau dari belakang. Sebagai contoh:

A Toyota: Race fast safe car: a Toyota

Tulis fungsi utama dalam C++ yang membaca satu baris teks dan meletakkan setiap huruf dalam tindanan dan baris gilir, kemudian mengesahkan sama ada teks adalah palindrom. Gunakan fungsi-fungsi asas tindanan dan baris gilir yang sedia ada.

(8/100)

- (b) *Deque* adalah baris gilir dua hujung, yang membolehkan penambahan dan penghapusan di kedua-dua hujung. Anggap ia akan dilaksanakan sebagai senarai berpaut ganda dua seperti yang dinyatakan di bawah:



```

template <class Type>
struct nodeType
{
    Type info;
    nodeType<Type> *next;
    nodeType<Type> *prev;
};

template<class Type>
class linkedDequeType
{
public:
    //All functions for Deque including ...
    void addDequeFront(const Type& newItem);
    //Add newItem in the front node
    void addDequeRear(const Type& newItem);
    //Add newItem in the back node
    void deleteDequeFront ( );
    //Delete an item from the front node
    void deleteDequeRear( );
    //Delete an item from the back node
    ~linkedDequeType();
    //destructor

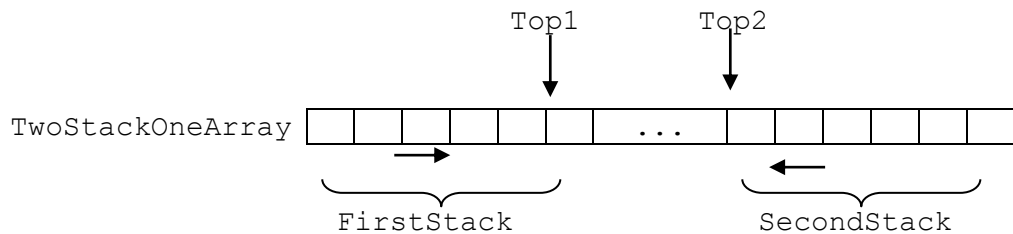
private:
    nodeType<Type> *dequeFront;
    //pointer to the front node
    nodeType<Type> *dequeRear;
    //pointer to the back node
};

```

Tulis dalam C++ definisi fungsi `deleteDequeRear ( )`;

(8/100)

- (c) Struktur storan yang menggunakan dua tindanan `FirstStack` dan `SecondStack` mempunyai jenis elemen yang sama; terdiri dari satu tatasusunan `TwoStackOneArray` dan dua penuding atas; `Top1` dan `Top2` yang membenarkan kedua-dua tindanan ini berkembang menghala arah satu sama lain; sebagaimana tergambar seperti di bawah:



- (i) Tulis dalam C++ definisi fungsi `push` dengan menggunakan tatasusunan dinamik. Fungsi ini menerima tindanan yang perlu untuk dioperasikan melalui parameter.
- (ii) Nyatakan keadaan bila fungsi `push` gagal?

(7/100)

- (d) Berikut merupakan senarai nod-nod bagi pepohon perduaan dalam 2 tertib yang berasingan:

tertib awalan: 49 28 18 11 19 40 32 44 83 71 69 72 97 92 99  
 tertib sisipan: 11 18 19 28 32 40 44 49 69 71 72 83 92 97 99

- (i) Lukis gambar rajah pepohon perduaan yang asal.
- (ii) Senaraikan nod tertib akhiran.
- (iii) Nyatakan penyusunan tertib yang mana yang selalu melawat:
- nod paling kiri dahulu?
  - nod paling kanan yang terakhir?

(7/100)