UNIVERSITI SAINS MALAYSIA

Second Semester Examination
[Peperiksaan Semester Kedua]

Academic Session 2007/2008
[Sidang Akademik 2007/2008]

April 2008

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

CPT103 – Data Structures & Programming Paradigms
[Struktur Data & Paradigma Pengaturcaraan]

Duration : 2 hours
[Masa : 2 jam]

INSTRUCTIONS TO CANDIDATE:
[ARAHAN KEPADA CALON:]

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

  *[Sila pastikan bahawa kertas peperiksaan ini mengandungi **EMPAT** soalan di dalam **TIGA 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.]

1. (a) Write C++ statements that multipliy a matrix, a[ ] [ ], and a vector, x[ ], giving a vector, b[ ] as the result. ([a][x] = [b]).

Example:

$$\begin{array}{cccc} [a] & [x] & & [b] \\ \begin{pmatrix} 2 & 4 & 5 \\ 9 & 8 & 4 \end{pmatrix} & \begin{bmatrix} 2 \\ 5 \\ 3 \end{bmatrix} & = & \begin{pmatrix} 39 \\ 70 \end{pmatrix} \end{array}$$

We get [b] =

$$\begin{array}{l} 2*2+4*5+5*3 = \\ 9*2+8*5+4*3 = \end{array} \begin{bmatrix} 39 \\ 70 \end{bmatrix}$$

(32/100)

(b) Explain the differences between `struct` and `class`.

(8/100)

(c) Consider the following statements:

```
struct nameType
{
   string first;
   string last;
};

struct dateType
{
   int month;
   int day;
   int year;
};

struct infoType
{
   nameType name;
   dateType birthday;
   int noShower[7];      //number of shower taken in one day for  a week
};

infoType person;
infoType desaHarapan[60];
```

(i) Write C++ statements to assign your name to `person`.

(8/100)

(ii) Write C++ statements to assign *Mat Clean* as the 20th name to `desaHarapan` occupant.

(8/100)

(iii) Write C++ statements to find the average number of times you showered in a week.

(16/100)

(iv) Write C++ statements to calculate total amount of water in liters used in a week by all the occupants in Desa Harapan who were born in April. Assume 5 liters are used on each shower.

(28/100)

2. (a) A point in the *x*-*y* plane is represented by its *x*-coordinate and *y*-coordinate.

Design a **class,** `PointType`, that can store and process a point in the *x*-*y* plane. Write the member functions to perform operations on a `point::` print the point, setting the coordinates of the point, printing the coordinates of the point, returning the *x*-coordinate and returning the *y*-coordinate.

(28/100)

(b) Every circle has a center and a radius. Given the radius, we can determine the circle's area and circumference. Given the center, we can determine its position in the x-y plane. The center of a circle is a point in the *x*-*y* plane.

Design a **class,** `CircleType`, that can store the radius and center of the circle. Because the center is a point in the *x*-*y* plane and the class to capture the properties of a point has been designed in Question 2(a), you must derive the **class** `CircleType` from the the **class** `PointType`.

Perform the usual operations on a `circle::` setting the radius, printing the radius, calculating and printing the area and circumference.

(36/100)

(c) Write a function **template** `smaller` to determine the smaller of two items.

(12/100)

(d) (i) What is a **friend** function?

(4/100)

(ii) Suppose that the `operator` << is to be overloaded for a user-defined **class** `mystery`. Why must << be overloaded as a **friend** function?

(8/100)

(iii) Consider the following recursive function:

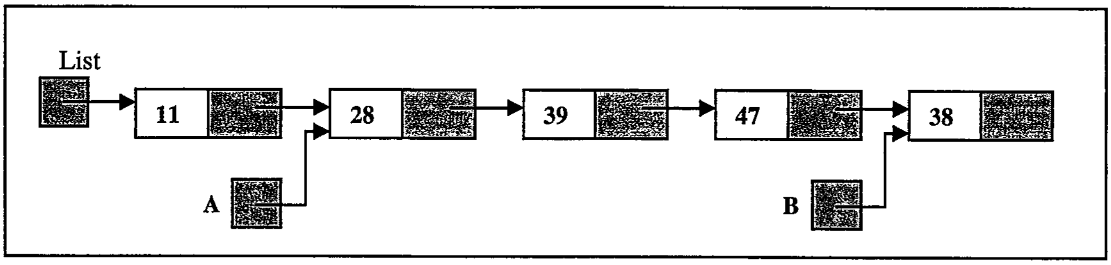
```
void  funcRec(int u, int v)
{
   if (u == 0)
      cout << v;
   else if (u == 1)
      cout << static_cast <char> (static_cast <int>(v) + 1);
   else
      funcRec(u - 1, v);
}
```

Identify the base case, the general case, and what is the output of the following statement?

```
funcRec( 5, 'A');
```

(12/100)

3.



Linked list for Question 3(a) through 3(b)(i).

(a) Based on the linked list diagram shown above, write C++ statements to do the following:

(i) Make **List** point to an empty list.

(ii) Create and insert the node with *info 15* after the node containing *info 47.*

(ii) Delete the node with *info 28.* Also, deallocate the memory occupied by this node.

(36/100)

(b) Determine the validity of the following C++ statements. If the statements are valid, show the output. If they are invalid, explain why?

(i)
```
P = list;

while (p != NULL)
cout << p->info << " ";
p = p->link;
cout << endl;
```

*Please note that the next 2 question were not related to the above diagram (shown on page 4).

(ii)
```
nodeType list;
nodeType ptr;

list = new nodeType;
list->info = 10;
list->link = NULL;
ptr = new nodeType;
ptr->info = 13;
ptr->link = list;
list = ptr;
ptr = new nodeType;
ptr->info = 18;
ptr->link = list->link;
list->link = ptr;
cout << list->info << ", " << ptr->info << ", ";
ptr = ptr->link;
cout << ptr->info << endl;
```

(iii)
```
stackType<int> stack;
int x, y;

x = 4;
y = 2;
stack.push(6);
stack.push(x);
stack.push(x + 1);
y = stack.top();
stack.pop();
stack.push(x + y);
x = stack.top();
stack.pop();

cout << "x = " << x << endl;
```

(40/100)

(c) Consider the following statements:

```
stackType<int> stack;
int x;
```

Suppose that the input is:

```
14  45  34  23  10  5  -999
```

Show what is output by the following segment of code:

```
stack.push(5);

cin >> x;

while (x != -999)
{
   if (x % 2 == 0)
   {
         if    (!stack.isFullStack())
                stack.push(x);
   }
   else
         cout << "x = " << x << endl;
   cin >> x;
}

cout << "Stack Elements: ";

while (!stack.isEmptyStack())
{
   cout << " " << stack.top();
   stack.pop();
}
cout << endl;
```

(24/100)

4. (a) What do the initially empty queues Q and T "looklike" after the following sequence of operations?

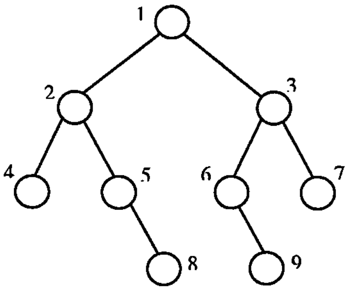
```
Q.QueueInsert(1, Success)
Q.QueueInsert(2, Success)
T.QueueInsert(3, Success)
T.QueueInsert(4, Success)
Q.QueueDelete(Success)
T.GetQueueFront(QueueFront, Success)
Q.QueueInsert(QueueFront, Success)
Q.QueueInsert(5, Success)
T.QueueDelete(QueueFront, Success)
T.QueueInsert(6,Success)
```

(20/100)

(b) Write a function; template, `reverse queue`, that takes as a parameter a queue object and uses a stack object, to reverse the elements of the queue.

(40/100)

(c) Consider the binary search tree:



The numbers simply label for nodes so that you can reference them; they do not indicate the contents of the nodes.

(i) Which node must contain the inorder successor of the value in the root? Explain.

(ii) In what order will an inorder traversal visit the nodes of this tree? Indicate this order by listing the labels of the nodes in the order that they are visited.

(20/100)

(d) Build a binary search tree using the following values:

62 40 20 15 32 102 54 47 155 133 112

(20/100)

KERTAS SOALAN DALAM VERSI BAHASA MALAYSIA

1. (a) Tulis kenyataan C++ yang mendarabkan satu matriks, a[ ] [ ], dan satu vektor, x[ ], untuk menghasilkan satu vektor, b[ ]. ([a][x] = [b]).

Contoh:

$$\begin{array}{cccc} [a] & [x] & & [b] \\ \begin{pmatrix} 2 & 4 & 5 \\ 9 & 8 & 4 \end{pmatrix} & \begin{bmatrix} 2 \\ 5 \\ 3 \end{bmatrix} & = & \begin{bmatrix} 39 \\ 70 \end{bmatrix} \end{array}$$

Kita dapat [b] =

$$\begin{array}{l} 2*2+4*5+5*3 = \\ 9*2+8*5+4*3 = \end{array} \begin{bmatrix} 39 \\ 70 \end{bmatrix}$$

(32/100)

(b) Terangkan perbezaan antara `struct` dan `class`.

(8/100)

(c) Pertimbangkan kenyataan berikut:

```
struct nameType
{
   string first;
   string last;
};

struct dateType
{
   int month;
   int day;
   int year;
};

struct infoType
{
   nameType name;
   dateType birthday;
   int noShower[7];      //bilangan mandi sehari untuk tempoh seminggu
};

infoType person;
infoType desaHarapan[60];
```

(i) Tulis kenyataan C++ yang umpukkan nama anda kepada `person`.

(8/100)

(ii) Tulis kenyataan C++ yang umpukkan *Mat Clean* sebagai nama ke- 20 penghuni `desaHarapan`.

(8/100)

(iii) Tulis kenyataan C++ yang akan mencari purata kekerapan anda mandi dalam seminggu.

(16/100)

(iv) Tulis kenyataan C++ yang mengira jumlah liter air yang digunakan dalam tempoh masa seminggu oleh semua penghuni Desa Harapan yang lahir dalam bulan April. Anggapkan 5 liter air digunakan untuk setiap mandian.

(28/100)

2. (a) Suatu titik dalam dimensi *x-y* diwakili dengan koordinat *x* dan koordinat *y*.

Reka bentuk satu **class,** `PointType`, yang akan menyimpan dan memproses satu titik dalam dimensi *x-y*. Tulis fungsi-fungsi ahli untuk melaksanakan operasi-operasi ke atas sesuatu titik, iaitu:: menunjukkan titik berkenaan, mensetkan koordinat-koordinat titik berkenaan, mencetakkan koordinat-koordinat sesuatu titik, mengembalikan koordinat-*x* dan koordinat-*y*.

(28/100)

(b) Setiap bulatan mempunyai satu pusat dan satu jejari. Diberikan satu jejari, kita boleh menentukan luas bulatan dan lilitannya. Pusat sesuatu bulatan adalah satu titik dalam dimensi *x-y*.

Reka bentuk satu **class,** `CircleType`, yang boleh menyimpan jejari dan pusat sesuatu bulatan. Oleh kerana pusat adalah satu titik dalam dimensi *x-y* dan **class** untuk mendapatkan ciri-ciri sesuatu titik itu telah direkabentuk di dalam Soalan 2(a), terbitkan **class** `CircleType` daripada **class** `PointType`.

Laksanakan operasi-operasi biasa ke atas bulatan, iaitu:: mensetkan jejari, mencetak jejari, mengira dan mencetak luas dan ukurlilit.

(36/100)

(c) Tulis satu **template** fungsi `smaller` untuk menentukan unsur terkecil daripada dua benda.

(12/100)

(d) (i) Apakah fungsi **friend**?

(4/100)

(ii) Andaikan bahawa `operator <<` diterlebihmuatkan untuk satu **class** takrifan pengguna `mystery`. Mengapa << mesti diterlebihmutakan sebagai satu fungsi **friend**?

(8/100)

(iii) Pertimbangkan fungsi rekursif berikut:

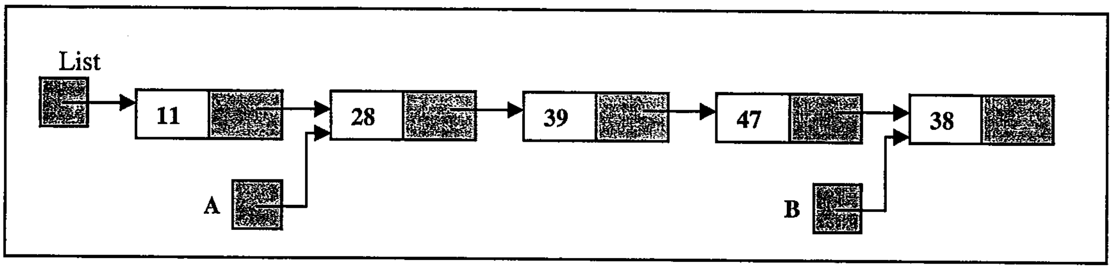
```
void  funcRec(int u, int v)
{
   if (u == 0)
      cout << v;
   else if (u == 1)
      cout << static_cast <char> (static_cast <int>(v) + 1);
   else
      funcRec(u - 1, v);
}
```

Kenalpasti kes dasar, kes umum, dan apakah output kenyataan berikut?

```
funcRec( 5, 'A');
```

(12/100)

3.



Gambar rajah senarai berpaut adalah untuk Soalan 3(a) hingga 3(b)(i) sahaja.

(a) Berdasarkan gambar rajah senarai berpaut yang ditunjukkan di atas, tuliskan kenyataan-kenyataan C++ untuk melakukan arahan berikut:

(i) Jadikan **List** menunjuk kepada senarai yang kosong.

(ii) Mewujudkan dan menyelitkan nod dengan *maklumat 15* selepas nod yang mengandungi *maklumat 47.*

(ii) Memadamkan nod *maklumat 28.* Juga arahan untuk menghapuskan ingatan yang diduduki oleh nod tersebut.

(36/100)

(b) Tentukan kesahihan kenyataan-kenyataan C++ berikut. Jika kenyataan-kenyataan adalah sah, tunjukkan outputnya dan jika ianya tidak sah, terangkan mengapa.

(i)
```
P = list;

while (p != NULL)
cout << p->info << " ";
p = p->link;
cout << endl;
```

*Soalan seterusnya tiada kaitan dengan gambar rajah senarai berpaut (di muka surat 9).

(ii)
```
nodeType list;
nodeType ptr;

list = new nodeType;
list->info = 10;
list->link = NULL;
ptr = new nodeType;
ptr->info = 13;
ptr->link = list;
list = ptr;
ptr = new nodeType;
ptr->info = 18;
ptr->link = list->link;
list->link = ptr;
cout << list->info << ", " << ptr->info << ", ";
ptr = ptr->link;
cout << ptr->info << endl;
```

(iii)
```
stackType<int> stack;
int x, y;

x = 4;
y = 2;
stack.push(6);
stack.push(x);
stack.push(x + 1);
y = stack.top();
stack.pop();
stack.push(x + y);
x = stack.top();
stack.pop();

cout << "x = " << x << endl;
```

(40/100)

(c) Pertimbangkan kenyataan-kenyataan berikut:

```
stackType<int> stack;
int x;
```

Anggap input adalah:

```
14  45  34  23  10  5  -999
```

Tunjukkan apakah hasil output bagi segmen kod yang berikut:

```
stack.push(5);

cin >> x;

while (x != -999)
{
   if (x % 2 == 0)
   {
         if    (!stack.isFullStack())
               stack.push(x);
   }
   else
         cout << "x = " << x << endl;
   cin >> x;
}

cout << "Stack Elements: ";

while (!stack.isEmptyStack())
{
   cout << "  " << stack.top();
   stack.pop();
}
cout << endl;
```

(24/100)

4. (a) Apakah kandungan giliran `Q` dan `T` (pada asalnya kosong) selepas semua jujukan operasi berikut?

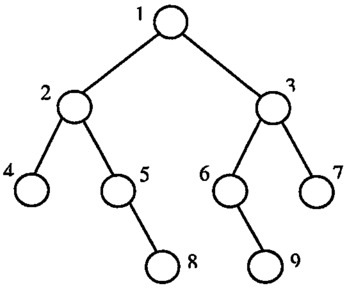
```
Q.QueueInsert(1, Success)
Q.QueueInsert(2, Success)
T.QueueInsert(3, Success)
T.QueueInsert(4, Success)
Q.QueueDelete(Success)
T.GetQueueFront(QueueFront, Success)
Q.QueueInsert(QueueFront, Success)
Q.QueueInsert(5, Success)
T.QueueDelete(QueueFront, Success)
T.QueueInsert(6,Success)
```

(20/100)

(b) Tulis satu templat fungsi; bernama `reverse queue`, yang menerima satu objek giliran sebagai parameter dan menggunakan satu objek tindihan untuk menterbalikkan elemen-elemen dalam giliran.

(40/100)

(c) Pertimbangkan pohon gelintaran perduaan berikut:



Nombor-nombor yang digunakan hanyalah sebagai label kepada nod supaya mudah anda merujuk. Nombor pada nod bukanlah kandungan nod.

(i) Nod yang mana mengandungi penurut tertib sisipan nilai dalam akar. Terangkan.

(ii) Dalam tertib apakah penyusuran tertib sisipan melawat nod-nod pohon ini? Tentukan tertib dengan menyenaraikan label pada nod-nod yang dilawati.

(20/100)

(d) Bina pohon gelintaran perduaan dengan menggunakan nilai berikut:

62 40 20 15 32 102 54 47 155 133 112

(20/100)

- oooOooo -