UNIVERSITI SAINS MALAYSIA

Second Semester Examination
2015/2016 Academic Session

June 2016

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 **TWELVE** printed pages before you begin the examination.

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

...2/-

1. (a) Consider the following declarations:

Pertimbangkan pengisytiharan berikut:

```
class bagType
{
public:
   void set(string, double, double, double);
   void print() const;
   string getStyle() const;
   double getPrice() const;
   void get(double, double, double, double);
   bagType();
   bagType(string, double, double, double, double);
private:
   string style;
   double l;
   double w;
   double h;
   double price;
};

bagType newBag; // variable declaration
```

(i) How many members does class `bagType` have?

Berapa banyak ahli-ahli bagi kelas `bagType`?

(ii) How many private members does class `bagType` have?

Berapa banyak ahli-ahli persendirian bagi kelas `bagType`?

(iii) How many constructors does class `bagType` have?

Berapa banyak pembina-pembina bagi kelas `bagType`?

(iv) How many constant functions does class `bagType` have?

Berapa banyak fungsi-fungsi pemalar bagi kelas `bagType`?

(v) Which constructor is used to initialize the object `newbag`?

Pembina yang manakah digunakan untuk memberi nilai awal pada objek `newbag`?

(5/100)

(b) (i) Make the necessary changes to the code given below to indicate that the class `LargeAnimal` is derived from the base class `ZooAnimal`, and it is a public inheritance.

Buat perubahan-perubahan yang perlu bagi kod yang diberi di bawah untuk menunjukkan kelas `LargeAnimal` diwarisi daripada kelas asas `ZooAnimal`, dan ianya secara pewarisan awam.

```
class ZooAnimal
{
private:
   char* name;
   int cageNumber;
   int weightData;
   int weight;
public:
   ZooAnimal(char*, int, int, int);
   ~ZooAnimal () { delete [] name; };
   void changeWeight (int);
   char* reptName ();
   int reptWeight ();
   int daysSinceLastWeighed (int);
};
class LargeAnimal
{
private:
   char* species;
   float cageMinimumVolume;
public:
   LargeAnimal(char*, int, int, int, float);
   ~LargeAnimal () { delete [] species; };
   float reptCageMinimumVolume ();
};
```

(5/100)

...4/-

(ii) Make the necessary changes below to indicate that the class `LargeAnimal` is derived from the base classes `ZooAnimal` and `Mammal` (in that order), and it is a public inheritance.

Buat perubahan-perubahan yang perlu bagi kod yang diberi di bawah untuk menunjukkan bahawa kelas `LargeAnimal` diwarisi daripada kelas-kelas asas `ZooAnimal` dan `Mammal` (dalam susunan yang diberi), dan ianya secara pewarisan awam.

```
class ZooAnimal
{
protected:
   char* name;
   int cageNumber;
   int weightData;
   int weight;
public:
   ZooAnimal(char*, int, int, int);
   ~ZooAnimal () { delete [] name; };
   void changeWeight (int);
   char* reptName ();
   int reptWeight ();
   int daysSinceLastWeighed (int);
};

class Mammal
{
protected:
   float minimumVolume;
   int minimumWeight;
public:
   Mammal(float, int);
   ~Mammal (){};
   float reptminimumVolume ();
   int reptminimumWeight ();
};

class LargeAnimal
{
protected:
   char* species;
   float cageMinimumVolume;
public:
   LargeAnimal(char*, int, int, int, float, float, int);
   ~LargeAnimal () { delete [] species; };
   float reptcageMinimumVolume ();
};
```

(5/100)

...5/-

(c) (i) Write a C++ code segment to declare a class that contains an array of integers.

Tulis satu segmen kod C++ untuk mengisytiharkan satu kelas yang mengandungi satu tatasusunan integer.

(ii) Initialize the integer array in 1(c)(i) in the constructor of the class.

Nilai awalkan tatasusunan dalam 1(c)(i) dalam pembina kelas.

(iii) Then declare two friend class functions to the class declared in 1(c)(i) to find the largest and the smallest integers in the array.

Kemudian isytiharkan dua fungsi rakan kelas kepada kelas dalam 1(c)(i) untuk mencari integer terbesar dan terkecil di dalam tatasusunan.

(iv) Declare destructor that sets all of the elements in the array to 0 value.

Isytihar pemusnah yang menetapkan semua unsur-unsur dalam tatasusunan kepada nilai 0.

(9/100)

(d) (i) Find the error(s) in the following code:

Cari kesalahan(kesalahan) dalam kod di bawah:

```
class secret                                   //Line 1
{                                              //Line 2
public:                                        //Line 3
   secret operator>=(secret);                  //Line 4
   secret();                                   //Line 5
   secret(int, int);                           //Line 6
private:                                       //Line 7
   int a;                                      //Line 8
   int b;                                      //Line 9
};                                             //Line 10
```

...6/-

(ii) Find the error(s) in the following code:

Cari kesalahan(kesalahan) dalam kod di bawah:

```
class discover                                     //Line 1
{                                                  //Line 2
public:                                            //Line 3
   discover operator+(const discover& a,
                      const discover& b);          //Line 4
        //Returns the object containing the
        //num of the corresponding members
        //of the objects a and b

   discover();                                     //Line 5
   discover(int, int);                             //Line 6
private:                                           //Line 7
   int float;                                      //Line 8
   int second;                                     //Line 9
};                                                 //Line 10
```

(iii) Find the error(s) in the following code:

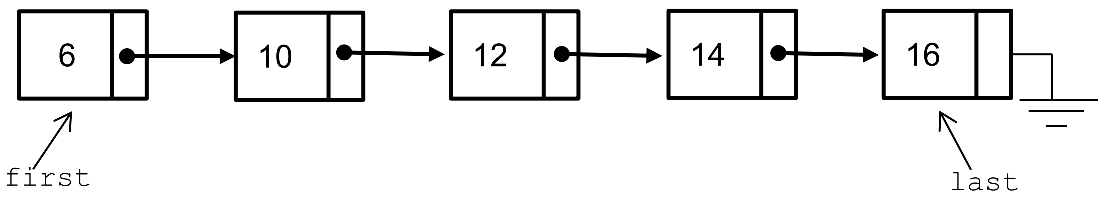
Cari kesalahan(kesalahan) dalam kod di bawah:

```
class mystery                                         //Line 1
{                                                     //Line 2
   friend mystery operator+(const mystery& a,
                      const mystery& b) const;  //Line 3
        //Return true if object a is less than
        //object b; otherwise it return false
   .
   .
   .
private:                                              //Line 4
   double r:                                          //Line 5
};                                                    //Line 6
```

(6/100)

2. (a) The following diagram represents nodes which stores baby's age in months in ascending order:

Gambar rajah berikut mewakili nod-nod yang menyimpan umur bayi dalam susunan menaik:



(i) Write C++ code to declare a structure named `baby` and declare `first` and `last` as pointers typed `baby`.

Tulis kod C++ untuk mengisytiharkan struktur bernama `baby` serta mengisytiharkan `first` dan `last` sebagai penuding berjenis `baby`.

(ii) Explain what make the problem of the following C++ code. Write the correct C++ code.

Terangkan apa yang menyebabkan masalah dalam kod C++ berikut. Tulis kod C++ yang betul.

(A)
```
newNode = new baby;
newNode->data=8;
first->next = newNode;
newNode->next=first->next;
```

(B)
```
baby *temp;
temp = first->next->next;
delete temp;
first->next->next=temp->next;
```

(iii) Rewrite the structure declaration in 2(a)(i) so that each node can be linked to previous node and the next node. Write the C++ code to print the baby's age in descending order.

Tulis semula pengisytiharan struktur supaya setiap nod boleh dipautkan ke nod sebelum dan nod seterusnya. Tulis kod C++ untuk mencetak umur bayi dalam susunan menurun.

(20/100)

...8/-

(b) The following is the incomplete abstract class definition as ADT:

Berikut adalah definisi kelas abstrak sebagai ADT yang tidak lengkap:

```
template <class Type>
class listType
   {
   public:
      const listType<Type> & operator=(const listType<Type>&);
      bool isEmptyList() const;
      void display();
      int length() const;
      void destroyList(); //function to delete all the nodes
      Type front() const;
      Type back() const;
      void copyData(const listType <Type> &,int);
      void insert_inOrder(_1_);
      void deleteNode (Type& id);
      (_2_)// default constructor
      (_3_)// default destructor

   protected:
      (_4_)// to declare pointer named first
      (_5_)// to declared pointer named last
   };
```

(i) Write the incomplete statements for question 2(b) (1),(2),(3),(4) and (5) above. Assume that the structure name to declare the node is `nodeType`.

Tulis peryataan-peryataan yang tidak lengkap untuk soalan 2(b) (1), (2),(3),(4) dan (5) di atas. Anggapkan nama struktur yang mengisytiharkan nod adalah `nodeType`.

(ii) Write C++ code to complete the function `copyData(const listType <Type> &,int)`. This function receives two parameters which are an address of an object and an integer value. This function will copy the content in of the existing linked list which is bigger than the received integer value into a new linked list.

Tulis kod C++ untuk melengkapkan fungsi `copyData(const listType <Type> &,int)`. Fungsi ini menerima dua parameter iaitu alamat objek dan satu nilai integer. Fungsi ini akan menyalin kandungan senarai berpaut sedia ada yang nilainya lebih besar daripada nilai integer yang diterima ke dalam senarai berpaut yang baru.

(15/100)

...9/-

3. (a) Write a C++ function `reverseStack()` to reverse the content of a stack using recursion method. Stack is represented using a linked list. Given the following structure of stack node, function prototypes, the main program and sample output:

Tulis fungsi C++ `reverseStack()` *untuk terbalikkan kandungan tindanan menggunakan kaedah rekursi. Tindanan diwakili menggunakan senarai berpaut. Diberi struktur nod tindanan, prototaip-prototaip fungsi, atur cara utama dan contoh output seperti berikut:*

```
//structure of a stack node
struct sNode
{
    char data;
    struct sNode *next;
};

//Function Prototypes
void push(struct sNode* top_ref, int new_data);
//Push an item into stack
int pop(struct sNode* top_ref);
//Pop an item from stack
bool isEmpty(struct sNode* top);
//Check if stack is empty
void print(struct sNode* top);
//Print the content of stack

//Main program
int main()
{
    struct sNode *s = NULL;
    push(&s, 4);
    push(&s, 3);
    push(&s, 2);
    push(&s, 1);

    cout<<"\n Original Stack ";
    print(s);
    reverseStack(&s);
    cout<<"\n Reversed Stack ";
    print(s);
    return 0;
}
```

Sample output:
Contoh output:

```
Original Stack
1  2  3  4
Reversed Stack
4  3  2  1
```

(12/100)

...10/-

(b) Priority Queue is different from normal queue because instead of being ‘first-in-first-out’ FIFO, items come out in order by priority. When inserting an item into priority queue, that item is sorted according to its priority, and ensures that the front item in the priority queue has the highest priority.

Baris gilir keutamaan adalah berbeza dari baris gilir biasa kerana selain dari ‘masuk dulu keluar dahulu' FIFO, item keluar mengikut keutamaannya. Apabila memasukkan item ke dalam baris gilir keutamaan, item tersebut disusun mengikut keutamaan, dan memastikan bahawa item yang berada di depan baris gilir keutamaan mempunyai keutamaan yang tertinggi.

Given the node structure and class definition `Priority_Queue`:

Di beri struktur nod dan definisi kelas `Priority_Queue`:

```
//Node Declaration
struct node
{
   int priority;
   int info;
   struct node *link;
};

//Class Priority Queue
class Priority_Queue
{
   private:
      node *front;
   public:
      Priority_Queue()
      {
         front = NULL;
      }
      //Insert into Priority Queue
      void insert(int, int);

      //Delete from Priority Queue
      void del();
```

...11/-

```
    //Print Priority Queue
    void display()
    {
       node *ptr;
       ptr = front;
       if (front == NULL)
           cout<<"Queue is empty\n";
       else
       {
           cout<<"Queue is :\n";
           cout<<"Priority Item\n";
           while(ptr != NULL)
           {
               cout<<ptr->priority<<endl;
               cout<<ptr->info<<endl;
               ptr = ptr->link;
           }
       }
    }
 };
```

(i) Write the function definition for `insert(int item, int priorityN).`

Tulis definisi fungsi untuk `insert(int item, int priorityN).`

(ii) Write the function definition for `del().`

Tulis definisi fungsi untuk `del().`

(iii) State the condition when `del()` function fails.

Nyatakan keadaan apabila fungsi `del()` gagal.

(10/100)

(c) Complete the following C++ main program that reverses the content of a queue using stack operations.

Lengkapkan atur cara utama C++ berikut untuk terbalikkan kandungan baris gilir dengan menggunakan operasi-operasi tindanan.

```
#include<iostream>
//add std library for stack and queue
#include<stack>
#include<queue>
using namespace std;

int main()
{
    //define queue and stack ADT
    queue <int> Q;
    stack <int> st;
```

...12/-

```
    //Push integer values 1 to 10 into Q

    //Take out elements from Q and push elements to st

    //Pop elements from st and push into Q

    //Print queue elements in reverse order

    return 0;
}
```

(i) Write the statements to push integer values from 1 to 10 into `Q`.

Tulis pernyataan-pernyataan untuk memasukkan nilai integer dari 1 hingga 10 ke dalam `Q`.

(ii) Write the statements to take out elements from `Q` and push into `st`.

Tulis pernyataan-pernyataan untuk mengeluarkan elemen dari `Q` dan masukkan ke dalam `st`.

(iii) Write the statements to pop elements from `st` and push into `Q`.

Tulis pernyataan-pernyataan untuk mengeluarkan elemen dari `st` dan masukkan ke dalam `Q`.

(iv) Write the statements to print queue elements in reverse order.

Tulis pernyataan-pernyataan untuk mencetak elemen baris gilir dalam susunan terbalik.

(13/100)

- oooOooo -