



Second Semester Examination
2017/2018 Academic Session

May/June 2018

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

Duration : 2 hours
(Masa : 2 jam)

Please ensure that this examination paper contains FOURTEEN (14) printed pages before you begin the examination.

[Sila pastikan bahawa kertas peperiksaan ini mengandungi EMPAT BELAS (14) muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]

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

[Arahan: Jawab **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. Given the following class definition:

Diberi definisi kelas berikut:

```
class basic
{
    private:
        int X;
    protected:
        int Y;
        void setX(int a){X=a;}
    public:
        void setY(int b){Y=b;}
        int getX(){return x;}
        void print();//display the value of X and Y
        basic(){X=Y=1;}
        basic(int a, int b){X=a;Y=b;}
        ~basic();
};
```

- (a) Suppose class Understand is derived from the class basic. Indicate whether each members of the class basic is **private**, **protected**, **public** or **inaccessible** for the following class declaration.

*Andaikan class Understand diterbitkan dari class basic. Tunjukkan sama ada setiap ahli class basic adalah **private**, **protected**, **public** atau **inaccessible** untuk pengisytiharan kelas berikut:*

class Understand:private basic		class Understand:public basic	
X		X	
Y		Y	
setX(..)		setX(..)	
setY(..)		setY(..)	
getX()		getX()	

(10/100)

...3/-

- (b) Given the following class definition.

Diberi definisi kelas berikut:

```
class Understand: public basic
private:
    int Z;
public:
    void setZ(int aa){Z=aa;}
    int getZ(){return Z;}
    Understand(){Z = 2;}
    Understand(int a, int b, int c);
    int CalVal ();
    void print();
    ~Understand();
};
```

- (i) How many members are accessible in the class Understand?

Berapa banyak ahli boleh dicapai di dalam kelas class Understand?

(4/100)

- (ii) Write a complete method/function for alternate constructor.
[Parameter - 1st and 2nd for base class and 3rd for derived class].

Tulis kaedah/fungsi untuk pembina alternative bagi class Understand.

[Parameter - 1 dan 2 untuk kelas asas dan 3 untuk kelas terbitan]

(6/100)

- (iii) Write a complete method/function definition for CalVal(). The CalVal() method/function should compute and return the value of $(X+Y)^Z$.

Tulis definisi kaedah/fungsi untuk CalVal(). Kaedah/fungsi CalVal() perlu mengira dan memulangkan nilai $(X+Y)^Z$.

(8/100)

...4/-

- (c) Given the following main() function:

Diberi fungsi main() berikut:

```
int main () {  
    Understand obj1;  
    Understand obj2(2,2,3);  
    cout << obj1.CalVal();  
    cout << obj2.CalVal();  
    obj1.setX(1);  
    obj1.setZ(4);  
    obj2.setY(2);  
    cout << obj1.CalVal();  
    cout << obj2.CalVal();  
    system("pause");  
}
```

- (i) How many separate instances of the **X, Y, Z** members exist? Justify your answer.

*Berapa banyak ketikaan berasingan bagi ahli-ahli **X, Y, Z**? Justifikasikan jawapan anda.*

(4/100)

- (ii) Give the output and explain how you get the output.

Beri output dan terang bagaimana anda dapat output.

(8/100)

- (d) Design a class named Person and its two (2) derived classes named Student and Employee. Make Lecturer a derived class of Employee. A person has a first and last name. A student has matric number, courses enrolled and status that are first-year, second-year, third-year, or final-year. Each student can take at most six (6) courses each semester. Define a class named Course that contains course name, course code, and course unit. An employee has salary, and date-hired. Define a class named Date that contains day, month, and year. A lecturer has consultation fees. Use appropriate constructors, destructor, composition, function overrides, function overloads, overload operator function operator> and friend function whenever necessary.

...5/-

Reka bentuk kelas bernama *Person* dan dua (2) kelas terbitan yang dinamakan *Student* dan *Employee*. Buatlah *Lecturer* sebagai kelas terbitan *Employee*. Seseorang mempunyai nama pertama dan nama akhir. Pelajar mempunyai nombor matriks, kursus yang didaftarkan dan status iaitu tahun-pertama, tahun-kedua, tahun-ketiga, atau tahun-akhir. Setiap pelajar boleh mengambil paling banyak enam (6) kursus setiap semester. Takrif kelas bernama *Course* yang mengandungi nama kursus, kod kursus, dan unit kursus. Seorang pekerja mempunyai gaji, dan tarikh lantikan. Takrif satu kelas bernama *Date* yang mengandungi hari, bulan, dan tahun. Pensyarah mempunyai yuran perundangan. Gunakan pembina yang bersesuaian, destruktur, komposisi, fungsi tanpa hirau, fungsi beban lampau, pengendali fungsi operator beban> dan fungsi kawan sekiranya perlu.

- (i) Draw the UML diagram for the classes together with their relations.

Lukis gambarajah UML untuk kelas-kelas termasuk hubungan antara mereka.

(15/100)

- (ii) Write the class named *Student* and the following methods:

- *setStudentInfo* to set student's information, including name, matric number, courses enrolled, and status.
- *print* to print student's name, matric number, status, total unit, and for each course enrolled, the course code, course name, and course unit.
- *getTotalUnit* to return the total number of unit courses enrolled.
- *getStatus* to return student's status.

Tulis kelas yang diberi nama Student dan kaedah berikut:

- *setStudentInfo* untuk menetapkan maklumat pelajar, termasuk nama, nombor matrik, kursus yang didaftarkan, dan status.
- *print* untuk mencetak nama pelajar, nombor matrik, status, jumlah unit, dan untuk setiap kursus yang didaftarkan, kod kursus, nama kursus, dan unit kursus.
- *getTotalUnit* untuk mengembalikan jumlah unit kursus yang didaftarkan.
- *getStatus* untuk memulangkan status pelajar.

(15/100)

...6/-

(iii) Write the class named *Lecturer* and the following methods:

- *setLecInfo* to set lecturer's information.
- *print* to print lecturer's name, salary and date hired.
- *getsalary* to return salary based on the sum of pay rate multiply with hours worked and consultation fees multiply with hours worked.

Tulis kelas yang bernama Lecturer dan kaedah berikut:

- *setLecInfo* untuk menetapkan maklumat pensyarah.
- *print* untuk mencetak nama pensyarah, gaji dan tarikh lantikan.
- *getsalary* untuk mengembalikan gaji berdasarkan jumlah kadar gaji darab dengan jam bekerja dan yuran perundingan darab dengan jam bekerja.

(15/100)

(iv) Write main function that creates **two (2)** persons for each student and lecturer. Display all information of the lecturer who has higher pay. Display all information of the student who has higher total number of unit courses enrolled.

Tulis fungsi main yang mencipta dua (2) orang untuk setiap pelajar dan pensyarah. Paparkan semua maklumat pensyarah yang mempunyai gaji yang lebih tinggi. Paparkan semua maklumat pelajar yang mempunyai bilangan unit kursus mendaftar yang lebih tinggi.

(15/100)

2. (a) Describe the following terms. Give your examples:

Jelaskan istilah berikut. Berikan contoh supaya penerangan menjadi lebih jelas:

(i) Deep copy.

Salinan dalam.

(ii) Shallow copy.

Salinan cetek.

(10/100)

...7/-

...8/-

SULIT

- (b) Complete the following program to print the words in a reverse order using a dynamic two-dimensional array. You are not allowed to create new variable.

Lengkapkan program berikut agar ia dapat mencetak perkataan yang dimasukkan secara terbalik menggunakan tatasusunan dinamik dua dimensi. Anda tidak dibenarkan mewujudkan pemboleh ubah baru.

```
int main()
{
    int row, column = 25;
    cout << "How many words will you enter? ";
    cin >> row;

    (i) Declare dynamic 2D array name wordList
    (ii) Initialised dynamic array wordList
    int i;
    for (i=0; i < row; i++)
        wordList[i] = new char [column];

    for (i = 0; i < row; i++)
    {
        cout << "Enter word " << i+1 << ": ";
        cin >> wordList[i];
    }

    cout << "Print in reverse: " << endl;
    for (i = (iii) )
    {
        cout << wordList[i] << endl;
    }

    // release dynamically allocated memory
    (iv) //loop
        (v) release memory// each array of characters

    (vi) release memory// the array of pointers
```

(30/100)

- (c) Given the following code:

Diberi kod berikut:

```
struct nodeType{
    int info;
    nodeType* link;
};
```

The following code is supposed to insert a new node at the end of a linked list. However, there is an error in the code.

Atur cara C++ berikut sepatutnya memasukkan satu node baru di penghujung satu senarai berpaut. Walau bagaimanapun, terdapat satu kesalahan dalam atur cara tersebut.

- (i) Identify and explain what is the problem.

Kenalpasti apakah kesalahan tersebut dan huraikan.

- (ii) Modify the code to make it work.

Ubahsuai atur cara itu untuk membolehkan atur cara itu dilaksanakan dengan betul.

```
//insert from the end of linked list
void linkedListType::insertLast(const int &n)
{
    nodeType *newNode = new nodeType;
    newNode->info = n;
    newNode->link = NULL;

    if(first != NULL)
    {
        first = newNode;
        last = newNode;
        count++;
    }
    else
    {
        last->link = newNode;
        last = newNode;
        count++;
    }
}
```

(15/100)

...10/-

- (d) Given you have a private member: `nodeType *first`, `nodeType *last` and `int count`. And also given that you have all the data abstraction for linked list, write a function for class `LinkedListType` named `sortLinkedList()`.

*Anda mempunyai ahli (pemboleh ubah) private: `nodeType *first`, `nodeType *last` dan `int count`. Diberi juga anda mempunyai semua fungsi dalam data abstrak untuk senarai berpaut, tuliskan satu fungsi untuk kelas `LinkedListType` bernama `sortLinkedList()`.*

(20/100)

- (e) T is a binary tree of height 4. Explain with example, what is the largest number of nodes that T can have? What is the smallest number?

T ialah pokok perduaan dengan ketinggian 4. Jelaskan dengan contoh, apakah bilangan terbesar nod yang dimiliki oleh T ? Apakah nombor terkecil?

(10/100)

- (f) Given the following binary tree orders:

- Preorder : N O P Q R S T U V W X Y Z
- Inorder : P R Q S O N U W V X T Z Y

- (i) Draw the binary tree.
(ii) List the nodes in postorder sequence.

Diberi senarai turutan nod seperti berikut:

- *Tertib awalan : N O P Q R S T U V W X Y Z*
- *Tertib sisipan : P R Q S O N U W V X T Z Y*

- (i) *Lukiskan pepohon penduaan yang terhasil.*
(ii) *Senaraikan turutan nod dalam tertib akhiran.*

(15/100)

3. (a) Given the following abstract class that defines a few operations of a stack and queue as an ADT.

Diberi kelas abstrak yang mentakrifkan beberapa operasi tindakan dan baris gilir secara ADT.

Stack	Queue
<pre> class stackType : public stackADT { public: void initializeStackType() void push(int); void pop(int); int top(); stackType(); stackType(const stackType&); ~stackType(); private: nodeType *stackTop; }; </pre>	<pre> class queueType : public queueADT { public: void initializeQueueType() void addQueue(int); void deleteQueue(); int front(); int back(); queueType(); queueType(const queueType&); ~queueType(); private: nodeType *queueFront; nodeType *queueRear; }; </pre>

- (i) Write the function definitions of pop of class stackType.

Tulis definisi fungsi pop kelas stackType.

(15/100)

- (ii) Write the function definitions of addQueue of class queueType.

Tulis definisi fungsi addQueue kelas queueType.

(15/100)

- (b) Given the following statements, state the content of stack s starting from the top.

Diberi penyata-penyata berikut, nyatakan kandungan tindanan s bermula dari bahagian atas.

```
int x;  
int tmp = 10;  
int num = 2;  
Stack s;  
s.push(num);  
for(int i = 0; i < 5; i++) {  
    x = s.top();  
    tmp = x + i;  
    s.push(tmp);  
}  
s.pop();
```

(10/100)

- (c) Given the following statements, state the content of queue q1 starting from the front.

Diberi penyata-penyata berikut, nyata kandungan baris gilir q1 bermula dari bahagian depan.

```
int x;  
int tmp = 3;  
int num = 4;  
q1.addQueue(num);  
for(int i = 0; i < 12; i=i+2) {  
    x = q1.back();  
    tmp = x + i;  
    q1.addQueue(tmp);  
}  
q1.deleteQueue();
```

(10/100)

- (d) Postfix notation is an arithmetic expression notation in which the operands appear before the operators. No parentheses are required in postfix notation. The evaluation of an arithmetic expression in postfix notation involves a data structure to store the element of the expression. The expression is scanned from left to right. For each element of the expression, the following rules (algorithm) are applied:

Notasi postfix adalah notasi ungkapan aritmetik di mana operan ditulis sebelum operasi. Kurungan tidak diperlukan dalam notasi postfix. Penilaian sesuatu ungkapan aritmetik dalam bentuk notasi postfix melibatkan struktur data untuk menyimpan elemen ungkapan. Ungkapan diimbis dari kiri ke kanan. Setiap elemen ungkapan mengikut peraturan-peraturan (algoritma) berikut:

- Step 1: if the element is a variable or constant, store it on the data structure.
 Step 2: if the element is an operator, two most recent elements are taken from the data structure, perform the operation and store the result into the data structure.

Langkah 1: jika elemen ialah pemboleh ubah atau pemalar, simpan ke dalam struktur data.

Langkah 2: jika elemen adalah operasi, dua elemen yang terakhir dikeluarkan dari struktur data and jalankan operasi tersebut dan simpan jawapan ke dalam struktur data.

The following examples shows an arithmetic expression in postfix notation.

Contoh-contoh di bawah menunjukkan ungkapan aritmetik dalam notasi postfix.

Arithmetic expression	Postfix notation
$(a+b) \times c$	$ab+c \times$
$(a-b)/(d \times e+c)$	$ab-de \times c+ /$

- (i) State the most appropriate data structure to be used for the above application.

Nyatakan struktur data yang paling sesuai untuk aplikasi di atas.

(5/100)

...14/-

- (ii) Write the program segment to implement the postfix conversion algorithm using ADT of the chosen data structure. Assume that a string variable called *expression* is used to store the postfix notation expression and a function to convert a character to number is available and is called *charToNum*.

*Tulis segmen program untuk melaksanakan algoritma penukaran postfix menggunakan ADT struktur data pilihan. Anggap satu pemboleh ubah aksara yang dipanggil *expression* digunakan untuk menyimpan ungkapan notasi postfix dan fungsi untuk menukar aksara ke nombor telah diberi yang dipanggil *charToNum*.*

(20/100)

- (e) (i) Given the following code:

Diberikan kod seperti berikut:

```
void my_recursive_function()
{
    my_recursive_function();
}
int main()
{
    my_recursive_function();
    return 0;
}
```

What will happen when above code is executed? Explain your answer.

Apa yang akan berlaku apabila kod di atas dilaksanakan? Terangkan jawapan anda.

(5/100)

- (ii) You improve the code in question 3(e)(i) as follows:

Anda memperbaiki kod pada soalan 3(e)(i) seperti berikut:

```
void my_recursive_function(int n)
{
    if(n == 0)
        return;

    cout << " "<< n;
    my_recursive_function(n-1);
}
int main()
{
    my_recursive_function(10);
    return 0;
}
```

What is the output of the code (if any)? How many times is the recursive function called when the following code is executed?

Apakah output kod (jika ada)? Nyatakan bilangan berapa kali fungsi rekursi dipanggil apabila kod tersebut dilaksanakan?

(10/100)

- (f) The Fibonacci sequence is 0, 1, 1, 2, 3, 5, 8, 13 such that the first two terms are 0 and 1, and each term thereafter is defined recursively as the sum of the two preceding terms; that is,

Turutan Fibonacci ialah 0,1,1,2,3,5,8,13... kedua-dua istilah pertama dan kedua adalah 0 dan 1, dan setiap istilah ditakrifkan secara rekursi sebagai jumlah dari dua istilah sebelumnya; dengan diberikan,

$$\text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2)$$

Write a recursive function that returns the nth number in a Fibonacci sequence when n is passed to the function as an argument. For example, when n=7, the function returns the 7th number in the sequence, which is 8.

Tulis fungsi rekursi yang mengembalikan nombor n dalam urutan Fibonacci apabila n diberikan kepada fungsi sebagai hujah. Sebagai contoh, apabila n=7, fungsi tersebut akan menunjukkan nombor ke-7 dalam urutan, iaitu nombor 8.

(10/100)

- 0000000 -