Second Semester Examination
2016/2017 Academic Session

June 2017

CPT111/CPM111 – Principles of Programming
[Prinsip Pengaturcaraan]

Duration : 2 hours
[Masa : 2 jam]

INSTRUCTIONS TO CANDIDATE:
[ARAHAN KEPADA CALON:]

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

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

- Answer **ALL** questions related to programming in C++ programming language.

  *[Jawab **SEMUA** soalan yang berkenaan dengan pengaturcaraan dalam bahasa pengaturcaraan C++.]*

- Questions ONE (1) and TWO (2) are to be answered in one answer booklet while question THREE (3) and FOUR (4) are to be answered in another answer booklet. You are required to answer in two different answer booklet.

  [Soalan SATU (1) dan DUA (2) dijawab dalam buku jawapan yang sama manakala soalan TIGA (3) dan EMPAT (4) dijawab dalam buku jawapan yang lain. Anda perlu menjawab dalam dua buku jawapan yang berlainan.]

- 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) Explain briefly in terms of their capabilities the key difference between the following.

Terangkan dengan ringkas dari segi kemampuan perbezaan utama antara berikut.

(i) a compiler and an interpreter.

pengkompil dengan penterjemah.

(ii) a while loop and a do while loop.

gelung while dengan gelung do while.

(iii) a for loop and a while loop.

gelung for dengan gelung while.

(iv) a switch and a if else.

switch dengan if else.

(12/100)

(b) What are the output of these programs?

Apakah output bagi atur cara-atur cara berikut?

(i)

```
#include <iostream>
using namespace std;
int main()
{
   int  const  p = 5;
   cout << ++p;
   return 0;
}
```

(ii)

```
#include <iostream>
using namespace std;
int main()
{
   int num1 = 5;
   int num2 = 3;
   int num3 = 2;
   num1 = num2++;
   num2 = --num3;
   cout << num1 << num2 << num3;
   return 0;
}
```

(iii)
```
#include <cstdlib>
#include <ctime>
#include <iostream>
using namespace std;
int main()
{
   srand((unsigned)time(0));
   int ran;
   for (int i = 0; i < 2; i++)
   {
      ran = (rand() % 10) + 1;
      cout << ran;
   }
}
```

(iv)
```
#include <iostream>
using namespace std;
int main()
{
   int a = 5, b = 6, c;
   c = (a > b) ? a : b;
   cout << c;
   return 0;
}
```

(12/100)

2. (a) What are the output of these program segments when x is 9 and y is 11 and when x is 11 and y is 9?

Apakah output bagi cebisan atur cara-atur cara berikut apabila x adalah 9 dan y adalah 11 dan apabila x adalah 11 and y adalah 9?

(i)
```
if ( x < 10 )
if ( y > 10 )
cout << "*****" << endl;
else
cout << "#####" << endl;
cout << "$$$$$" << endl;
```

(ii)
```
if ( x < 10 )
{
if ( y > 10 )
cout << "*****" << endl;
}
else
{
cout << "#####" << endl;
cout << "$$$$$" << endl;
}
```

(6/100)

(b) Write a program to display a simple menu selection using switch and input user selection for the following sample output.

Tulis satu atur cara untuk memaparkan satu pilihan menu yang mudah dengan menggunakan pilihan switch dan menginput pilihan pengguna untuk sampel output berikut.

Output examples:

Contoh Output

```
Menu                               Menu
========                           ========
A - Append                         A - Append
M - Modify                         M - Modify
D - Delete                         D - Delete
X - Exit                           X - Exit
Enter selection: a                 Enter selection: d
To append a record                 To delete a record
Press any key to continue . . .    Press any key to continue . . .

Menu                               Menu
========                           ========
A - Append                         A - Append
M - Modify                         M - Modify
D - Delete                         D - Delete
X - Exit                           X - Exit
Enter selection: A                 Enter selection: x
To append a record                 To exit the menu
Press any key to continue . . .    Press any key to continue . . .

Menu                               Menu
========                           ========
A - Append                         A - Append
M - Modify                         M - Modify
D - Delete                         D - Delete
X - Exit                           X - Exit
Enter selection: M                 Enter selection: T
To modify a record                 Invalid selection
Press any key to continue . . .    Press any key to continue . . .
```

(10/100)

(c) Write a program to print all Armstrong numbers between 1 and 500. If sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number.

For example, 153 = (1 * 1 * 1) + (5 * 5 * 5) + (3 * 3 * 3)

Tulis satu atur cara untuk mencetak semua nombor Armstrong antara 1 dan 500. Jika jumlah ganda tiga setiap angka nombor adalah sama dengan bilangan itu sendiri, nombor itu dipanggil nombor Armstrong.

*Sebagai contoh, 153 = (1 * 1 * 1) + (5 * 5 * 5) + (3 * 3 * 3)*

(10/100)

3. (a) What are the output of the C++ statements in (i) and (ii)?

Apakah output bagi kenyataan-kenyataan C++ dalam (i) dan (ii)?

(i)

```
void sum (int a, int b, int& c);

int main()
{
   int x = 2, y =3;
   sum(x, y, y);
   cout << " x = " << x << " y = " << y;
   return 0;
}

void sum (int a, int b, int& c)
{
   a = b + c;
   b = a + c;
   c = a + b;
   cout << " a = " << a << " b = " << b << " c = " << c;
}
```

(ii) Assume that the memory locations for x is 0x22fe3c, y is 0x22fe30, p is 0x22fe28, and q is 0x22fe29.

Anggap lokasi memori bagi x ialah 0x22fe3c*, y ialah* 0x22fe30*, p ialah* 0x22fe28*, dan q ialah* 0x22fe29*.*

```
int x;
int y;
int *p;
int *q;
p = &x;
q = &y;
x = 35;
y = 46;
p = q;
*p = 27;
cout << “ x = “ << x << “  y = “  << y << endl;
cout << “ *p = “ << *p << “  *q= “ << *q << endl;
cout << “ p = “ << p << endl;
```

(10/100)

(b) The factorial of a number is given by the equation:

```
factorial = 1 * 2 * 3... * n
```

For example, factorial of 3 = 3 * 2 * 1 = 6.

Faktorial bagi satu nombor diberi oleh formula:

```
factorial = 1 * 2 * 3... * n
```

*Contohnya, faktorial bagi 3 = 3 * 2 * 1 = 6.*

(i) Write the main algorithm (pseudocode) that prompts user to enter an integer. Then the factorial of that integer is computed and displayed on the screen.

Tulis algoritma utama (pseudokod) yang akan prom pengguna untuk memasukkan satu integer. Seterusnya, faktorial integer tersebut akan dikira dan dipapar di atas skrin.

(ii) Write a C++ program that uses a `factorial()` function to compute the factorial. Include the main function, function prototype and function definition. Store the factorial result in a variable of `int` data type.

Tulis atur cara C++ yang menggunakan fungsi untuk mengira faktorial. Sertakan fungsi utama, prototaip fungsi dan definisi fungsi. Simpan keputusan faktorial dalam pemboleh ubah jenis `int`*.*

(iii) What happens when user enters the following input values? Explain your answer.

- zero or negative number
- 100

Apakah yang akan berlaku apabila pengguna memasukkan nilai input yang berikut? Beri penjelasan bagi jawapan anda.

- *sifar atau nombor negatif*
- *100*

(iv) Modify the `factorial()` function so that the any negative integer will be changed to positive and factorial of any integer greater than 12 will return the value of 0. List additional built-in libraries you use that are not included in 3(b)(ii) (if any).

Ubah fungsi `factorial()` supaya nombor negative akan ditukar menjadi positif dan faktorial bagi integer lebih besar daripada 12 akan memulangkan nilai 0. Senaraikan sebarang perpustakaan tersedia C++ yang anda guna dan tidak termasuk dalam 3(b)(ii) (jika ada).

(15/100)

4. (a) Given the following C++ program:

Diberi atur cara C++ berikut:

```
#include <iostream>
using namespace std;
int main ()
{
   int list1[] = {1, 2, 4, 6, 7, 5, 3};
   int n, result = 0;
   int multiply;
   multiply = list1[2] * list1[5];
   for (n = 0; n < 8; n++)
   {
      result += list1[n];
   }
   cout << "multiply = " << multiply << endl;
   cout << "result = " << result << endl;
   return 0;
}
```

(i) How many components are there in array `list1[]`?

Berapakah komponen dalam tatasusunan `list1[]`?

(ii) What is the lowest and highest index of the array?

Apakah indeks tatasusunan terendah dan tertinggi?

(iii) What is the output of the program?

Apakah output bagi atur cara ini?

(5/100)

(b) Consider the outline of the program below and write appropriate C++ codes for the following questions:

Pertimbangkan garis panduan bagi atur cara di bawah dan tulis keratan atur cara C++ yang sesuai bagi soalan-soalan berikut:

```
#include <iostream>
#include <string>
#include <fstream>
#include <iomanip>
using namespace std;
const int SIZE = 5;
int  main()
{
     string firstName[SIZE];
     double weight[SIZE];
     double height[SIZE];
     double bmi[SIZE];
     ifstream infile;
     ofstream outfile;

     infile.open("students.txt");
     outfile.open("bmi.txt");
     ...
     ...
     infile.close();
     outfile.close();
     return 0;
}
```

(i) Initialize all components in `height[]` to 0.

Tetapkan nilai awalan semua komponen dalam `height[]` kepada 0.

(ii) Given the sample input file `"students.txt"` below. The first column shows the student first name, the second column shows weight in kilograms (`kg`), and the third column shows height in meters (`m`).

Diberi contoh fail input `"students.txt"` seperti di bawah. Lajur pertama menunjukkan nama pertama pelajar, lajur kedua menunjukkan berat dalam kilogram (`kg`), dan lajur ketiga menunjukkan tinggi dalam meter (`m`).

```
Andrew      70.0  1.75
Kathy       55.5  1.88
Miller      64.2  2.11
Gina        88.9  1.67
Desmond     90.7  1.98
```

Read data from `"students.txt"` into the three parallel one-dimensional arrays. Array `firstName[]` should store student first name, `weight[]` should store student weight in `kg`, and `height[]` should store student height in `m`.

Baca data daripada `"students.txt"` ke dalam tiga tatasusunan satu dimensi yang selari. Tatasusunan `firstName[]` perlu menyimpan nama pertama pelajar, `weight[]` perlu menyimpan berat palajar dalam `kg`, dan `height[]` perlu menyimpan tinggi pelajar dalam `m`.

(iii) Write a function definition to calculate the body mass index (BMI) of the students and store the BMI of each student in the `bmi[]` array. Name the function `computeBMI()`. BMI is calculated based on the following formula (weight in `kg` and height in `m`):

Tulis definisi fungsi untuk mengira indeks jisim badan (BMI) pelajar dan simpan BMI bagi setiap pelajar dalam tatasusunan `bmi[]`. Namakan fungsi ini `computeBMI()`. BMI dikira berdasarkan formula berikut (berat dalam `kg` dan tinggi dalam `m`):

BMI = weight / (height * height)

(iv) Write a function call in `main()` to execute the `computeBMI()` function. Pass the appropriate actual parameters to calculate BMI for each student.

Tulis panggilan fungsi dalam `main()` untuk memanggil fungsi `computeBMI()`. Serahkan parameter sebenar yang sesuai untuk mengira BMI bagi setiap pelajar.

(v) Write a function definition `averageBMI()` that calculates and returns the value average BMI of two students.

Tulis fungsi `averageBMI()` untuk mengira dan memulangkan nilai purata BMI bagi dua orang pelajar.

(vi) Write a function call in `main()` to compute the average BMI of the first and last students in the array. Display the first names of the two students and the average BMI value.

Tulis panggilan fungsi dalam `main()` untuk mengira purata BMI bagi pelajar pertama dan terakhir dalam tatasusunan. Paparkan nama pertama dua orang pelajar ini dan nilai purata BMI mereka.

(vii) Save the BMI for each student in the format below to a file designated by variable `outfile` (“bmi.txt”). Format your output to one decimal place.

Simpan BMI bagi setiap pelajar mengikut format di bawah ke dalam fail yang ditetapkan oleh pemboleh ubah `outfile` (“bmi.txt”). Format output kepada satu tempat perpuluhan.

```
Name        BMI
Andrew      22.9
Kathy       15.7
Miller      14.4
Gina        31.9
Desmond     23.1
```

(20/100)

- oooOooo -