
UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Pertama
Sidang Akademik 2001/2002

September 2001

CPT101/CAP101 – Prinsip-Prinsip Pengaturcaraan

Masa : 3 jam

ARAHAN KEPADA CALON:

- Sila pastikan bahawa kertas peperiksaan ini mengandungi **EMPAT** soalan di dalam **SEMBILAN** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.
 - Jawab **SEMUA** soalan dalam Bahasa Malaysia.
-

1. (a) Nyatakan sama ada kenyataan-kenyataan berikut BENAR atau PALSU.
 - (i) Komputeran pelayan/pelanggan adalah merupakan salah satu mod komputer yang wujud dalam persekitaran komputeran teragih.
 - (ii) Maklumat dalam ingatan bantu disimpan dalam unit logikal yang dipanggil rekod.
 - (iii) Penterjemah bahasa merupakan suatu perisian yang menukarkan atur cara yang ditulis dalam bahasa peringkat tinggi kepada bahasa mesin.
 - (iv) Algoritma hanya boleh diterangkan dengan menggunakan struktur pilihan dan ulangan sahaja.
 - (v) Pengujian dan Pengesahsahihan adalah langkah terakhir dalam proses pembangunan perisian.
 - (vi) Pengisytiharan pemboleh ubah boleh dilakukan di mana-mana tempat dalam badan fungsi C++.
 - (vii) Komen menyebabkan komputer mencetak teks selepas simbol // di atas skrin apabila atur cara dilaksanakan.
 - (viii) Terdapat tiga jenis ralat utama iaitu ralat sintaks, ralat masa-larian, dan ralat logik.
 - (ix) Pseudokod dan carta alir adalah dua teknik untuk mewakili algoritma.
 - (x) Keperluan input dan output perlu dikenalpasti pada peringkat spesifikasi keperluan semasa membangunkan atur cara yang bersistem. [20/100]

- (b) Terangkan dengan ringkas maksud dan perbezaan bagi setiap pasangan istilah berikut:
 - (i) Ingatan Capaian Rawak (RAM) dan Ingatan Baca Sahaja (ROM).
 - (ii) Pentafsir dan Pengkompil.
 - (iii) Bahasa Mesin dan Bahasa Himpunan. [30/100]

- (c) Sebuah syarikat mempunyai sistem pembayaran gaji mengikut kategori jawatan dan pengiraan pembayaran gaji dilakukan setiap minggu. Syarikat ini mempunyai ramai pekerja yang terdiri dari kategori jawatan penyelia (bergaji tetap), pekerja am (gaji mengikut jumlah jam berkerja), penjual (gaji tetap dan komisyen jualan) dan tukang (gaji mengikut jumlah penghasilan produk, seorang tukang hanya membuat satu jenis produk). Syarikat ini boleh mengeluarkan pelbagai jenis produk. Jadual di bawah menunjukkan perincian bayaran gaji untuk setiap kategori pekerja.

Kategori Jawatan	Bayaran Gaji Seminggu
Penyelia	Gaji tetap.
Pekerja am	Untuk 30 jam pertama, kadar tetap untuk setiap jam. Untuk jam selebihnya, gaji lebih masa pada kadar 1.5% daripada kadar tetap setiap jam.
Penjual	RM200 serta komisyen 5.5% daripada jumlah jualan seminggu.
Tukang	Kadar upah darab jumlah bilangan produk yang dihasilkan seminggu.

- (i) Senaraikan semua input dan rumusan yang diperlukan untuk mengira gaji mingguan bagi setiap kategori jawatan.
- (ii) Selesaikan masalah pengiraan gaji mingguan mengikut penghalusan berperingkat dengan membina suatu carta struktur.
- (iii) Tulis pseudokod untuk mengira gaji mingguan bagi setiap pekerja dalam syarikat ini mengikut kategori jawatan mereka.
- (iv) Tulis keratan atur cara dalam C++ bagi masalah pengiraan gaji mingguan dengan menggunakan sama ada struktur kawalan **if-else** atau **switch**.

[50/100]

2. (a) Atur cara berikut akan menukarkan waktu dalam bentuk 24 jam kepada bentuk 12 jam. Contohnya waktu 14:25 ditukar kepada 2:25 PM. Input akan dibaca dalam bentuk dua data integer (jam dan minit). Atur cara ini mengandungi tiga fungsi, satu untuk membaca input, satu untuk melakukan penukaran waktu dan satu lagi untuk mencetak output. Nilai AM dan PM disimpan sebagai satu nilai aksara, 'A' untuk AM dan 'P' untuk PM, oleh itu fungsi penukaran adalah berjenis aksara yang dipanggil secara rujukan. Fungsi `baca_data` dipanggil secara rujukan menggunakan pemboleh ubah penuding, manakala fungsi `cetak` dipanggil menggunakan panggilan secara nilai (Takrifan fungsi `cetak` diberikan di bawah). Lengkapkan atur cara ini dengan memberikan panggilan dan takrifan fungsi-fungsi yang digunakan.

```
#include <iostream.h>

// prototaip ketiga-tiga fungsi tersebut.

void baca_data(int *, int *);
char penukaran(int, int&);
void cetak(char, int, int, int);

void main()
{
    int jam24, jam12, minit;
    char am_pm;

    // berikan di sini panggilan bagi ketiga-tiga fungsi
    // menggunakan pemboleh ubah yang telah diisytihar di atas
}

// berikan di sini takrifan untuk fungsi baca_data dan fungsi
// penukaran

// takrifan fungsi cetak

void cetak(char AM_PM, int jam_asal, int jam_baru, int minit)
{
    cout << jam_asal << ":" << minit << " ditukar kepada ";
    cout << jam_baru << ":" << minit << ' ' << AM_PM << 'M';
    cout << endl;
}
```

[50/100]

- (b) (i) Tulis satu keratan atur cara menggunakan gelung **for** dan formatkan output menggunakan **setw**, bagi menghasilkan jadual seperti berikut:

-	0	1	2	3	4
0	0	-1	-2	-3	-4
1	1	0	-1	-2	-3
2	2	1	0	-1	-2
3	3	2	3	0	-1
4	4	3	2	1	0

(ii) Apakah output yang dihasilkan oleh keratan atur cara berikut:

```
x = -42;
do
{
    cout << x << endl;
    x = x - 3;
} while (x > 0);
cout << x << endl;
```

[25/100]

(c) Atur cara di bawah bertujuan untuk mengira markah purata setiap pelajar dan markah purata keseluruhan semua pelajar bagi tiga markah ujian. Data maklumat pelajar yang terdiri daripada nombor matrik dan tiga markah ujian dibaca daripada satu fail input luaran dan hasil pengiraan ditulis ke dalam satu fail output luaran.

Lengkapkan atur cara berikut dengan menulis kenyataan C++ yang bersesuaian di dalam ruang yang berlabel (i), (ii), (iii), (iv) dan (v).

```
#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>

void main()
{
    int nomatrik;
    int markah1, markah2, markah3, bil;
    float j_semua, mpurata, purata_semua;
    j_semua = 0;

    ofstream fail_cetak;
    ifstream fail_data;

    // buka kedua-dua fail di mana nama fail luaran adalah
    // "cetak.txt" untuk fail output dan
    // "data.txt" untuk fail input.

    (i)

    // uji sama ada kedua-dua fail ini dapat dibuka atau tidak,
    // jika gagal paparkan mesej "fail gagal dibuka".

    (ii)

    while (!fail_data.eof())
    {

    // baca data nomatrik dan tiga markah pelajar daripada
    // fail input

    (iii)
```

```

    if (fail_data.eof())
        break;
    bil++;
    mpurata = (markah1 + markah2 + markah3) / 3.0;
    j_semua = j_semua + mpurata;

// tulis nomatrik dan mpurata ( markah purata)
// ke dalam fail output

    (iv)
}

purata_semua = j_semua / bil;
fail_cetak << "purata keseluruhan " << purata_semua;

// tutup kedua-dua fail input dan output tersebut.

    (v)
}

```

[25/100]

3. Andaikan setiap permainan hoki melibatkan empat suku di antara dua pasukan. Tatasusunan berikut ditakrifkan untuk menyimpan gol yang dihasilkan di dalam setiap suku di dalam satu permainan.

```
int gol [2][4];
```

- (a) Tulis keratan atur cara yang akan menjana nilai di antara 0 dan 2 (termasuk 0 dan 2) sebagai gol bagi setiap suku untuk dua pasukan yang bertanding. Nilai tersebut perlu disimpan di dalam tatasusunan di atas.

[20/100]

- (b) Tulis keratan atur cara untuk mengira keputusan sesuatu permainan hoki, iaitu, menjumlahkan gol yang terkumpul di dalam tatasusunan di atas untuk kedua-dua pasukan yang bertanding. Jumlah untuk pasukan pertama disimpan di dalam pemboleh ubah bernama **p1** dan jumlah untuk pasukan kedua di dalam pemboleh ubah **p2**.

[20/100]

- (c) Pertandingan akhir hoki di antara UXM dan UYM akan melibatkan 5 permainan. Keputusan-keputusan permainan ini akan disimpan di dalam tatasusunan berikut:

```
int keputusan [5][2] = {{-1},{-1},{-1},{-1},{-1}};
```

Andaikan perlawanan pertama berakhir dengan keputusan 4 – 2. Maka kandungan tatasusunan keputusan ialah seperti berikut:

4	2
-1	0
-1	0
-1	0
-1	0

Maka jadual seperti berikut perlu dikeluarkan:

Permainan 1 (4 – 2)
 Permainan 2
 Permainan 3
 Permainan 4
 Permainan 5

Tulis keratan atur cara untuk menghasilkan jadual seperti di atas.

[30/100]

- (d) Tulis satu fungsi yang akan menerima satu keputusan permainan hoki. Keputusan berkenaan akan disemak dan jika didapati seri, maka gol tambahan akan dijana sehingga ada pemenang. (Gol tambahan perlu ditambah kepada gol yang sedia ada.) Nilai gol yang dijana di dalam masa tambahan sama ada 0 atau 1 sahaja. Keputusan permainan akan dipulangkan oleh fungsi tersebut.

[30/100]

4. (a) Berikut ialah satu pengisytiharan struktur bersarang:

```

struct RekodKemalangan{
    char tarikh[15];
    float kos;
    int bilkemalangan;
};
struct Lori{
    int purataperjalanan;
    int berat;
};
struct Kereta{
    int berat;
    int jumpenumpang;
    char jeniskereta[20];
};
union Kenderaan{
    Lori lori;
    Kereta kereta;
};
struct PelangganInsuran{
    char nama[30];
    int umur;
    char jantina;
    int jumlahkemalangan;
    RekodKemalangan RK[4];
    char jeniskenderaan;
    Kenderaan k;
    float premium;
};
PelangganInsuran pelanggan;

```

- (i) Lukis keadaan struktur bagi pemboleh ubah pelanggan.
- (ii) Tulis keratan atur cara C++ yang akan menentukan kos bayaran berdasarkan kepada jumlah kemalangan pelanggan.

Jumlah Kemalangan	Kos Bayaran
0 – 3	0
4 – 8	RM100.00 setiap satu
Lebih dari 8	RM50 setiap satu bagi 1 – 8 yang awal dan RM150 bagi setiap satu yang selebihnya

- (iii) Isytiharkan satu jenis terangkakan bernama **JK** yang mempunyai dua jenis kenderaan iaitu Lori dan Kereta.
- (iv) Tulis keratan atur cara C++ untuk mencetak maklumat kenderaan dengan menggantikan pemboleh ubah **jeniskenderaan** berjenis **char** kepada jenis terangkakan **JK** yang anda beri dalam 4(a)(iii) di atas.

[65/100]

- (b) Satu objek/rekod Pelajar mempunyai dua jenis data iaitu maklumat peribadi dan maklumat akademik. Maklumat peribadi mengandungi nama pelajar berjenis rentetan 20 aksara, nombor matrik pelajar dan alamat sahaja. Alamat menyimpan alamat Jalan, Bandar, Poskod dan Negeri. Maklumat akademik pelajar menyimpan markah ujian pelajar (maksimum tiga ujian bagi setiap pelajar), markah tugas pelajar (maksimum enam tugas) dan markah peperiksaan pelajar. Bagi setiap objek pelajar dua fungsi ditakrifkan iaitu fungsi **Bacadata** dan juga fungsi **KiraGred**.

Berdasarkan maklumat yang diberi di atas:

- (i) Tulis satu kelas Pelajar.
- (ii) Tulis satu pembina salinan yang akan menyalin data dari objek **Pelajar1** ke objek **Pelajar2** dengan anggapan yang **Pelajar2** telah terlebih dahulu dijanakan dan diberi nilai.

[35/100]