

UNIVERSITI MALAYA
UNIVERSITY OF MALAYA

PEPERIKSAAN IJAZAH SARJANA MUDA SAINS KOMPUTER / SARJANA MUDA
TEKNOLOGI MAKLUMAT
EXAMINATION FOR THE DEGREE OF BACHELOR OF COMPUTER SCIENCE / BACHELOR
OF INFORMATION TECHNOLOGY

SESI AKADEMIK 2015/2016 : SEMESTER I
ACADEMIC SESSION 2015/2016 : SEMESTER I

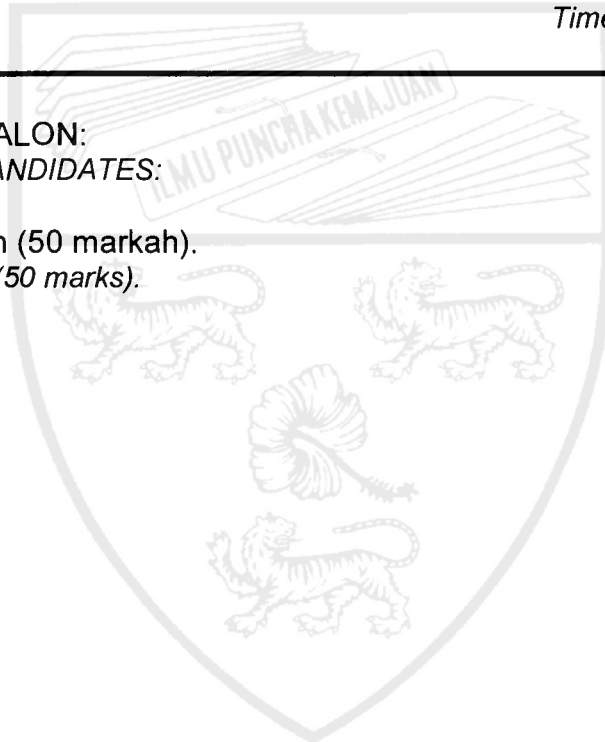
WIX1002 : Asas-Asas Pengaturcaraan
Fundamentals of Programming

Dis 2015/Jan 2016
Dec 2015/Jan 2016

Masa: 3 jam 30 minit
Time: 3 hours 30 minutes

ARAHAN KEPADA CALON:
INSTRUCTIONS TO CANDIDATES:

Jawab **SEMUA** soalan (50 markah).
Answer **ALL** questions (50 marks).



(Kertas soalan ini mengandungi 5 soalan dalam 8 halaman yang dicetak)
(This question paper consists of 5 questions on 8 printed pages)

1. Aturcara dalam fail **Q1.java** mengandungi banyak ralat. Betulkan kesemua ralat tersebut. (Salin fail **Q1.java** dari direktori akaun peperiksaan anda. Selepas aturcara dibetulkan, namakan fail tersebut sebagai [**matricNumberQ1.java**; contoh: **WEK150001Q1.java**] dan salin fail ini ke direktori akaun peperiksaan.)

*The program in the **Q1.java** file contains many errors. Correct all errors. (Copy the **Q1.java** file from your exam account directory. After the program has been corrected, name the file as [**matricNumberQ1.java**; example: **WEK150001Q1.java**] and copy this file to your exam account directory.)*

```
import java.util.Scanner;
// Filename: Q1.java
public class Q1 {
    public static void main(String[] args) {
        int[] num = {"34", "15", "12", "27", "74", "23"};
        int cnt=0;
        for (int i = 0; i < num.length(); i++ )
            if (isEven(num))
                cnt++;
        System.out.println("The number of even number is " + cnt);
        System.out.println("The sum of the array is " + getTotal(num[0]));
    }

    public boolean isEven(int a) {
        if (a%2=0)
            return true;
        else
            return false;
    }

    public static void getTotal(int[] a) {
        int total=0;
        for (int i = 0; i < a.length; i++ )
            total = a[i];
        return total;
    }
}
```

(4 markah/marks)

2. Tuliskan satu aturcara untuk permainan gunting, batu dan kertas. Aturcara tersebut akan membolehkan pengguna terus bermain sehingga sama ada pengguna atau komputer menang tiga kali. (Simpan aturcara tersebut dalam fail **Main.java**. Salin fail ini ke direktori akaun peperiksaan anda dan namakan semula sebagai **[matricNumberQ2.java]**; contoh: **WEK150001Q2.java**.)

*Write a program that plays the scissor, rock and paper game. The program will let the user continuously play until either the user or the computer wins three times. (Save the program in the **Main.java** file. Copy this file to your exam account directory and rename as **[matricNumberQ2.java]**; example: **WEK150001Q2.java**.)*

Contoh output:

Sample output:

```
Enter 1.Paper 2.Scissor 3.Rock: 1
Player : Paper ----- Computer : Scissor
Computer win 1 time(s)
Enter 1.Paper 2.Scissor 3.Rock: 3
Player : Rock ----- Computer : Scissor
Player win 1 time(s)
Enter 1.Paper 2.Scissor 3.Rock: 2
Player : Scissor ----- Computer : Paper
Player win 2 time(s)
Enter 1.Paper 2.Scissor 3.Rock: 2
Player : Scissor ----- Computer : Scissor
Enter 1.Paper 2.Scissor 3.Rock: 3
Player : Rock ----- Computer : Scissor
Player win 3 time(s)
Player wins the game.
```

(9 markah/marks)

3. Tuliskan satu aturcara untuk menjana **n** integer rawak dalam lingkungan **0-1000** dan simpan integer tersebut dalam satu tatasusunan. Kemudian, bina **EMPAT** kaedah untuk melaksanakan tindakan berikut.

- Paparkan setiap integer di dalam tatasusunan.
- Paparkan nombor maximum.
- Bundarkan setiap integer kepada puluh yang hampir dan paparkan.
- Paparkan setiap integer secara terbalik di dalam tatasusunan.

(Simpan aturcara tersebut dalam fail **Main.java**. Salin fail ini ke direktori akaun peperiksaan anda dan namakan semula sebagai [**matricNumberQ3.java**; contoh: **WEK150001Q3.java**].)

*Write a program to generate **n** random integers within **0 - 1000** and store the integers in an array. Then, create **FOUR methods** to perform the following actions.*

- *Display each of the integer in the array.*
- *Display the maximum number.*
- *Round each of the integer to the nearest tenth and display.*
- *Display each of the integer in reverse in the array.*

*(Save the program in the **Main.java** file. Copy this file to your exam account directory and rename as [**matricNumberQ3.java**; example: **WEK150001Q3.java**].)*

Contoh output:

Sample output:

```
Enter the number of random integer: 6
The random integer : 594 353 875 250 95 511
Maximum Number : 875
The approximation of the integer to the nearest tenth : 590 350 880 250 100 510
The random integer in reverse order: 495 353 578 52 59 115
```

(12 markah/marks)

4. Fail **Q4.txt** mengandungi satu senarai katalaluan. Tuliskan satu aturcara untuk menentukan sama ada katalaluan itu ialah katalaluan yang kukuh. Satu katalaluan yang kukuh mesti mempunyai yang berikut:

- Sekurang-kurangnya 8 aksara.
- Sekurang-kurangnya 1 aksara yang berharuf besar.
- Sekurang-kurangnya 1 aksara yang berharuf kecil.
- Sekurang-kurangnya 1 angka (0 - 9).
- Sekurang-kurangnya 1 aksara yang bukan abjad dan angka (aksara *space* tidak dikira).

(Salin fail **Q4.txt** dari direktori akaun peperiksaan anda. Simpan aturcara dalam fail **Main.java**. Salin fail **Main.java** ini ke akaun direktori peperiksaan anda dan namakan semula sebagai [**matricNumberQ4.java**; contoh: **WEK150001Q4.java**].)

*The **Q4.txt** file contains a list of password. Write a program to determine whether the password is strong password. A strong password must contain the followings:*

- *At least 8 characters.*
- *At least 1 uppercase character.*
- *At least 1 lowercase character.*
- *At least 1 digit (0 - 9).*
- *At least 1 non-alphanumeric characters (space character is not counted).*

*(Copy the **Q4.txt** file from your exam account directory. Save the program in **Main.java** file. Copy this **Main.java** file to your exam account directory and rename as [**matricNumberQ4.java**; example: **WEK150001Q4.java**].)*

Contoh output:

Sample output:

```
Strong password.
Strong password.
Not a strong password.
Not a strong password.
Not a strong password.
Not a strong password.
Strong password.
Not a strong password.
Not a strong password.
```

(10 markah/marks)

5. Kad kredit adalah satu kad bayaran yang dikeluarkan kepada pemegang kad sebagai kaedah pembayaran. Terdapat dua jenis kad kredit yang termuka iaitu kad rebat tunai dan kad mata ganjaran. Tulis satu aturcara untuk menentukan kad terbaik untuk digunakan.

A credit card is a payment card issued to cardholders as a method of payment. There are two popular types of credit card namely the cash rebate card and the point reward card. Write a program to determine the best card to use. The program will include the following:

- a) Reka bentuk satu kelas **CreditCard** yang terdiri daripada ahli berikut:
- Satu medan untuk nama pemegang kad.
 - Satu medan untuk nombor kad. (16 angka)
 - Satu medan untuk jenis kad. (Rebat Tunai / Mata Ganjaran)
 - Satu pembina yang mengandungi semua medan di atas.
 - Satu medan untuk jumlah ganjaran tunai dalam RM.
 - Satu kaedah *accessor* dan *mutator* untuk jumlah ganjaran tunai.
 - Satu kaedah *toString* yang memulangkan semua maklumat di atas.

*Design a **CreditCard** class that consists of the following members:*

- *A field for the name of the cardholder.*
- *A field for the number of the card. (16 digits)*
- *A field for the type of the card. (Cash Rebate / Point Reward)*
- *A constructor that contains all the above fields.*
- *A field for the total cash reward in RM.*
- *An accessor and mutator method for total reward.*
- *A toString method that returns the above information.*

(4 markah/marks)

- b) Kad rebat tunai menyediakan 8% rebat tunai untuk bahan api, 5% rebat tunai atas bayaran bil utiliti, 2% rebat tunai untuk barang runcit dan 0.2% rebat tunai untuk barang yang lain. Jika pemegang kad membelanjakan RM100 untuk bahan api, pemegang kad akan mendapat rebat tunai RM8. Reka bentuk satu kelas **Rebate** yang mewarisi kelas **CreditCard**. Kelas **Rebate** mesti mempunyai ahli berikut:

- Satu Pembina.
- Kaedah *getReward* yang mengira jumlah rebat tunai.

*A cash rebate card provides 8% cashback on fuel, 5% cashback on utility bill payment, 2% cashback for grocery and 0.2% cashback for other. If the cardholder spends RM100 for fuel, the cardholder will get RM8 cashback. Design a **Rebate** class that extends the **CreditCard** class. The **Rebate** class must have the following members:*

- *A constructor.*
- *A getReward method that computes the total cash rebate.*

(4 markah/marks)

- c) Kad mata ganjaran memberikan 1 mata untuk setiap perbelanjaan RM1. Jika pemegang kad membelanjakan RM42.75, pemegang kad akan mendapat 42 mata. Setiap mata boleh ditukar kepada wang tunai (1 mata = RM0.01). Mata kad ganjaran juga menyediakan dua kali ganda mata bagi apa-apa perbelanjaan pada hari Jumaat, tiga kali ganda mata bagi apa-apa perbelanjaan pada hari Sabtu, dan empat kali ganda mata bagi apa-apa perbelanjaan pada hari Ahad. Reka bentuk satu kelas **Point** yang mewarisi kelas **CreditCard**. Kelas **Point** mesti mempunyai ahli berikut:
- Satu Pembina.
 - Kaedah *getReward* yang mengira jumlah ganjaran tunai.

*A point reward card provides 1 point for each RM1 spend. If the cardholder spends RM42.75, the cardholder will get 42 points. Each point can be converted into cash. (1 point = RM0.01) The point reward card also provides double reward points for any spending on Friday, triple reward points for any spending on Saturday, and quadruple reward points for any spending on Sunday. Design a **Point** class that extends the **CreditCard** class. The **Point** class must have the following members:*

- A constructor
- A *getReward* method that computes the total cash reward.

(4 markah/marks)

- d) Bina satu kelas **Main** menggunakan kod di bawah untuk menguji aturcara tersebut. Lengkapkan aturcara tersebut untuk menentukan kad terbaik untuk digunakan berdasarkan perbelanjaan dalam jadual di bawah. Anggapkan bahawa hanya satu jenis kad boleh digunakan untuk semua perbelanjaan.

*Create a **Main** class using the code below to test the program. Complete the program to determine the best card to use based on the spending in table below. Assume that only one type of card can be used for all the spending.*

```
public static void main(String[] args) {
    Rebate r = new Rebate("John Lim", "1111222233334444");
    Point p = new Point("John Lim", "5555666677778888");
    // Complete the program
    // ...
}
```

No	Type	Day	Total (RM)
1.	Grocery	Saturday	124.80
2.	Other	Friday	64.60
3.	Fuel	Sunday	95.40
4.	Utility	Friday	100.00
5.	Other	Tuesday	220.00

Jadual 1: Jumlah Perbelanjaan
Table 1: Total Expenditure

(Salin fail **CreditCard.java**, **Rebate.java**, **Point.java** dan **Main.java** ke dalam direktori akaun peperiksaan anda.)

(Copy the **CreditCard.java**, **Rebate.java**, **Point.java** and **Main.java** files to your exam account directory.)

Contoh output:

Sample output:

```
Credit Card : John Lim (1111222233334444)
Card Type : Cash Rebate
Total Cash Reward : 15.6972
Credit Card : John Lim (5555666677778888)
Card Type : Point Reward
Total Cash Reward : 13.0
The best card is Cash Rebate Card
```

(3 markah/marks)

TAMAT
END