

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 2016/2017 : SEMESTER II  
ACADEMIC SESSION 2016/2017 : SEMESTER II

WIX1002 : Asas-Asas Pengaturcaraan  
*Fundamentals of Programming*

Jun 2017  
June 2017

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

ARAHAN KEPADA CALON:  
INSTRUCTIONS TO CANDIDATES:

Calon dikehendaki menjawab **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: **WIA160001Q1.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: **WIA160001Q1.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) {
        final int N;
        Scanner s = new Scanner(System.out);
        System.out.print("Enter a number : ");
        N = s.nextDouble();
        System.out.println("The first " + N + " triangular numbers");
        for(int i=1; i<N; i--)
            System.out.print(getTriangular(i) + " ");
        System.out.println();
    }

    public static double getTriangular(String n) {
        int sum;
        for(int i=1; i<=n; i++)
            sum+=1;
        return sum;
    }
}
```

(4 markah/marks)

2. Tuliskan sebuah aturcara yang meminta pengguna untuk memasukkan bilangan pasukan, nama pasukan dan markah pasukan berdasarkan jadual di bawah. Kemudian, paparkan nama pasukan dan markah masing-masing secara tertib menurun di mana markah pasukan tertinggi dahulu dipaparkan. (Simpan aturcara tersebut dalam fail **Main.java**. Salin fail ini ke direktori akaun peperiksaan anda dan namakan semula sebagai **[matricNumberQ2.java]**; contoh: **WIA160001Q2.java**).

*Write a program that asks a user to enter the number of teams, their team names, and their scores based on the table below. Then, display the team names and their scores in descending order with the highest team score first. (Save the program in the **Main.java** file. Copy this file to your exam account directory and rename as **[matricNumberQ2.java]**; example: **WIA160001Q2.java**).*

Team	Score
Norwich City	24
Newcastle United	42
Blackburn Rovers	12
Queens Park Rangers	20
Birmingham City	22

Contoh output:

Sample output:

```
Enter number of teams: 5
Enter team 1 name: Norwich City
Enter team 1 score: 24
Enter team 2 name: Newcastle United
Enter team 2 score: 42
Enter team 3 name: Blackburn Rovers
Enter team 3 score: 12
Enter team 4 name: Queens Park Rangers
Enter team 4 score: 20
Enter team 5 name: Birmingham City
Enter team 5 score: 22
List of team with the highest team score first
Newcastle United (42) | Norwich City (24) | Birmingham City (22) | Queens Park Rangers (20) | Blackburn Rovers (12) |
```

(8 markah/marks)

3. Tulis sebuah aturcara yang meminta pengguna memasukkan satu nilai integer,  $n$ . Daripada nilai integer ini, aturcara tersebut perlu menjana satu set integer secara rawak dalam lingkungan 0-10000. Simpan set integer tersebut dalam satu tatasusunan. Kemudian, bina EMPAT kaedah untuk melaksanakan tindakan berikut:
- Paparkan set integer di dalam tatasusunan.
  - Dapatkan nombor minimum.
  - Paparkan anggaran setiap integer kepada penghampiran seratus.
  - Paparkan setiap integer di dalam tatasusunan secara terbalik.

(Simpan aturcara tersebut dalam fail **Main.java**. Salin fail ini ke direktori akaun peperiksaan anda dan namakan semula sebagai [**matricNumberQ3.java**; contoh: **WIA160001Q3.java**].

*Write a program that asks a user to enter an integer value,  $n$ . Based on this integer value, the program should generate a set of random values within the range of 0-10000. Store the integer set in an array. Then, create FOUR methods to perform the following action:*

- *Display the set of integers in the array.*
- *Get the minimum number.*
- *Display the approximation of each integer value to the nearest hundred.*
- *Display each integer in the array in a reverse order.*

*(Save the program in the **Main.java** file. Copy this file to your exam account directory and rename as [**matricNumberQ3.java**; example: **WIA160001Q3.java**].*



Contoh output:

Sample output:

```
Enter the number of random integer: 5
The random integer : 5133 668 27 9652 1083
Minimum Number : 27
The approximation of the integer to the nearest hundred : 5100 700 0 9700 1100
The random integer in reverse order: 3315 866 72 2569 3801
```

(12 markah/marks)

4. *Zeller's congruence* digunakan untuk mengira hari dalam minggu. Diberi formula yang berikut:

$$h = \left( q + \left\lfloor \frac{13(m+1)}{5} \right\rfloor + K + \left\lfloor \frac{K}{4} \right\rfloor + \left\lfloor \frac{J}{4} \right\rfloor + 5J \right) \bmod 7$$

Simbol	Penerangan
h	hari dalam minggu (0: Sabtu, 1: Ahad, 2: Isnin ... 6:Jumaat)
q	hari dalam bulan
m	bulan (3: Mac, 4: April, ... 12:Disember, Januari dan Februari dikira sebagai 13 dan 14 pada tahun sebelumnya)
y	tahun
J	abad = tahun/100
K	tahun dalam abad = tahun modulus 100

Seorang ahli Sains ingin memaparkan hari minggu berdasarkan rekod yang disimpan dalam fail binari (**Q4.dat**). Setiap rekod dalam fail binari mengandungi q, m dan y secara teratur. Tulis satu program untuk membantu ahli Sains tersebut untuk memaparkan output. Program ini mengandungi kelas bernama **DayOfTheWeek**. Kelas tersebut mesti mengandungi perkara berikut:

- Pembolehubah *instance* untuk simbol-simbol di atas.
- Satu pembina yang mengandungi q, m dan y sebagai hujah.
- Satu kaedah untuk memastikan bahawa hari yang sah dalam bulan tertentu termasuk menentukan sama ada tahun ini adalah tahun lompat. Tahun lompat ialah tahun kalendar yang mengandungi satu hari tambahan pada bulan Februari.
- Satu kaedah untuk menentukan sama ada tahun ini adalah tahun lompat. Tahun lompat boleh dibahagikan dengan 4 tetapi tidak boleh dibahagi dengan 100. Walau bagaimanapun, tahun lompat boleh dibahagikan dengan 400.
- Satu kaedah untuk memaparkan hari dalam minggu.

Tulis sebuah kelas penguji untuk membaca semua rekod dari fail binari (**Q4.dat**) dan mengumpukkan nilai kepada objek **DayOfTheWeek**.

(Salin fail **Q4.dat** dari direktori akaun peperiksaan anda. Simpan kelas penguji dalam fail **Main.java**. Salin fail **DayOfTheWeek.java** dan **Main.java** ini ke akaun direktori peperiksaan anda and namakan **Main.java** semula sebagai **[matricNumberQ4.java]**; contoh: **WIA16000Q4.java**).

*Zeller's congruence is used to calculate the day of the week. Given the following formula:*

$$h = \left( q + \left\lfloor \frac{13(m+1)}{5} \right\rfloor + K + \left\lfloor \frac{K}{4} \right\rfloor + \left\lfloor \frac{J}{4} \right\rfloor + 5J \right) \bmod 7$$

Symbol	Description
$h$	day of the week (0: Saturday, 1: Sunday, 2: Monday ... 6:Friday)
$q$	day of the month
$m$	month (3: March, 4: April, ... 12:December, January and February are counted as 13 and 14 of the previous year)
$y$	year
$J$	century = year/100
$K$	year of the century = year modulus 100

A scientist would like to display the day of the week based on the records stored in a binary file (**Q4.dat**). Each record in the binary file contains  $q$ ,  $m$  and  $y$  in order. Write a program to help the scientist to display the output. The program contains a class named **DayOfTheWeek**. The class must contain the following:

- Instance variables for the symbols above.
- A constructor that contains  $q$ ,  $m$  and  $y$  as arguments.
- A method to ensure that the valid day of the month including determine whether the year is a leap year. A leap year is a calendar year containing one additional day in February.
- A method to determine whether the year is a leap year. A leap year is divisible by 4 but not by 100. In addition, a leap year is also divisible by 400.
- A method to display the day of the week.

Write a tester class to read all the records from the binary file (**Q4.dat**) and assign the value to the **DayOfTheWeek** object.

(Copy the **Q4.dat** file from your exam account directory. Save the tester class in **Main.java** file. Copy **DayOfTheWeek.java** and **Main.java** file to your exam account directory and rename the **Main.java** as **[matricNumberQ4.java]**; example: **WIA160001Q4.java**).

Contoh output:

Sample output:

```
25/3/2017 is on Saturday
29/2/2100 is invalid Input
19/1/2017 is on Thursday
31/9/2015 is invalid Input
31/5/2017 is on Wednesday
29/2/2016 is on Monday
```

(15 markah/marks)

5. Reka bentuk satu kelas **Staff** yang terdiri daripada ahli berikut:

- Satu medan untuk nama kakitangan.
- Satu medan untuk IC kakitangan.
- Satu pembina yang mengandungi semua medan di atas.
- Satu kaedah toString yang memaparkan nama kakitangan dan IC.

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

- A field for the name of the staff.
- A field for the IC of the staff.
- A constructor that contains all the above fields.
- A toString method that displays the staff's name and IC.

(3 markah/marks)

Reka bentuk satu kelas **PermanentStaff** yang mewarisi kelas **Staff**. Kelas **PermanentStaff** mempunyai ahli berikut:

- Satu medan untuk gred kakitangan.
- Satu pembina yang mempunyai hujah.
- Kaedah getSalary yang mengira gaji berdasarkan jadual di bawah.
- Satu kaedah toString yang memaparkan gred kakitangan dan gaji.

Design a **PermanentStaff** class that extends the **Staff** class. The **PermanentStaff** class has the following members:

- A field for the grade of the staff.
- A constructor with arguments.
- A getSalary method that computes salary based on the table below.
- A toString method that displays the staff's grade and salary.

Grade	Salary (RM)
EM01	7000
EM02	5000
EM03	3000
EM04	1000

**Salary Table**

(4 markah/marks)



Reka bentuk satu kelas **PartTimeStaff** yang mewarisi kelas **Staff**. Kelas **PartTimeStaff** mempunyai ahli berikut:

- Satu medan untuk bilangan jam bekerja (jenis integer).
- Satu pembina yang mempunyai hujah.
- Kaedah `getSalary` yang mengira gaji berdasarkan RM40 sejam.
- Satu kaedah `toString` yang memaparkan bilangan jam bekerja kakitangan dan gaji.

*Design a class **PartTimeStaff** that extends the **Staff** class. The **PartTimeStaff** class has the following members:*

- *A field for the number of working hours (integer type).*
- *A constructor with arguments.*
- *A `getSalary` method that computes salary based on RM40 per hour.*
- *A `toString` method that displays the staff's working hours and salary.*

(4 markah/marks)

Tulis sebuah kelas penguji Main menggunakan kod di bawah.

*Write a tester class Main using the code below.*

```
public static void main(String[] args) {
    PermanentStaff p1 = new PermanentStaff("Abdul Razak", "871001-14-1765",
"EM02" );
    PartTimeStaff p2 = new PartTimeStaff("Tengku Ali", "911223-08-3301", 35);
    System.out.println(p1.toString());
    System.out.println(p2.toString());
}
```

(Salin fail **Staff.java**, **PermanentStaff.java** dan **PartTimeStaff.java** ke dalam direktori akaun peperiksaan anda.)

*(Copy the **Staff.java**, **PermanentStaff.java** and **PartTimeStaff.java** files to your exam account directory.)*

Contoh output:

*Sample output:*

Full Name: Abdul Razak

IC: 871001-14-1765

Grade: EM02

Salary: RM 5000

Full Name: Tengku Ali

IC: 911223-08-3301

Number of working hours: 35

Salary: RM 1400

**TAMAT**  
**END**