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 2017/2018 : SEMESTER II
ACADEMIC SESSION 2017/2018 : SEMESTER II

WIX1002 : Asas-Asas Pengaturcaraan
Fundamentals of Programming

Mei/Jun 2018 Masa: 3 jam 30 minit
May/June 2018 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: **WIF170001Q1.java**] dan salin fail ini ke direktori akaun peperiksaan anda.)

   *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: **WIF170001Q1.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 = {"69", "15", "12", "27", "74", "23"};
    int cnt=0;
    for (int i = 0; i < num.length(); i++ )
       if (isOdd(num))
            cnt++;
    System.out.println("The number of odd number is " + cnt);
    System.out.println("The sum of the array is " + compSum(num[0]));
  }

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

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

} // end class
```

(10 markah/*marks*)

2. Tulis satu program yang menerima dari pengguna berat satu beg kopi dalam unit kilogram dan bilangan beg yang dijual. Paparkan jumlah harga jualan selepas cukai sebagai

   *totalPrice = unitWeight * numberOfBags * 5.99 ;*
   *totalPriceWithTax = totalPrice + totalPrice * 0.0725 ;*

   di mana 5.99 adalah kos satu kilogram dan 0.0725 adalah cukai jualan.

   Write a program that accepts from the user the unit weight of a bag of coffee in kilograms and the number of bags sold. Display the total price of the sale after tax, computed as

   *totalPrice = unitWeight * numberOfBags * 5.99 ;*
   *totalPriceWithTax = totalPrice + totalPrice * 0.0725 ;*

   where 5.99 is the cost per kilogram and 0.0725 is the sales tax.

   Contoh output:
   Sample output:

```
Enter the number of bags sold : 32<enter>
Enter the weight per bag (kilogram) : 5<enter>

  Price per kilogram:    $5.99
  Sales tax:              7.25%
  Total price after tax:  $ 1027.88
```

   (Simpan aturcara tersebut dalam fail **Q2Main.java**. Salin fail ini ke direktori akaun peperiksaan anda dan namakan semula sebagai [**matricNumberQ2.java**; contoh: **WIF170001Q2.java**].)

   *(Save the program in the **Q2Main.java** file. Copy this file to your exam account directory and rename as [**matricNumberQ2.java**; example: **WIF170001Q2.java**].)*

(5 markah/*marks*)

3. Fail teks, ***student.txt*** mengandungi 10 nama pelajar dan markah peperiksaan mereka. Sesiapa yang tidak menduduki peperiksaan tersebut akan mendapat -1 sebagai markah. Tulis satu program yang membaca dari fail teks tersebut, nama pelajar, ke dalam array jenis *String* bernama ***names*** dan markah yang berkaitan ke dalam satu array jenis *double* bernama ***marks***. Cari dalam array **marks** markah tertinggi dan terendah serta kenalpasti pelajar yang mendapat markah tertinggi atau terendah tersebut dari array ***names***. Kira bilangan pelajar yang tidak hadir ke peperiksaan tersebut. (Fail teks tersebut ada dalam direktori akaun peperiksaan anda.)

   *The text file, **student.txt,** contains 10 student names and their final exam marks. Those who did not sit for the exam will get -1 as their marks. Write a program that reads from the text file, the students' names, into a String array named **names** and the respective marks into an array of type double named **marks**. Search the **marks** array for the highest and lowest mark and identify the student who got the highest and lowest mark from the **names** array. Count the number of students who did not come for the exam. (The text file is available in your exam account directory.)*

   Contoh output:
   Sample output:

```
The student with the highest mark :  Frankenstein Jr 97
The student with the lowest mark :   Chewbacca Chewy 10
The number of students who were absent from the exam : 3
```

   (Simpan aturcara tersebut dalam fail **Q3Main.java**. Salin fail ini ke direktori akaun peperiksaan anda dan namakan semula sebagai **[matricNumberQ3.java**; contoh: **WIF170001Q3.java**].)

   *(Save the program in the **Q3Main.java** file. Copy this file to your exam account directory and rename as [**matricNumberQ3.java**; example: **WIF170001Q3.java**].)*

(15 markah/*marks*)

4. a) Takrif satu kelas bernama *LinearEquation* untuk satu sistem persamaan *linear* 2 x 2:

   Define a class named LinearEquation for a 2 x 2 system of linear equations:

   $ax + by = e$

   $cx + dy = f$ where $x = \frac{ed-bf}{ad-bc}$; $y = \frac{af-ec}{ad-bc}$

Kelas tersebut mengandungi:

The class contains:

- private data fields *a*, *b*, *c*, *d*, *e*, and *f*.
- a constructor with the arguments for *a*, *b*, *c*, *d*, *e*, and *f*.
- a method named **toString()** that returns the 2x2 linear equation.
- a method named **isSolvable()** that returns true if *ad* – *bc* is not 0.
- methods named **computeX()** and **computeY()** that returns the values for *x* and *y* as the solution for the equation.

(6 markah/*marks*)

b) Bina satu kelas **Q4Main** untuk menguji program. Ia meminta pengguna memasukkan nilai untuk *a, b, c, d, e,* dan *f* serta membina satu objek LinearEquation. Paparkan persamaan linear tersebut. Bergantung kepada nilai *ad-bc*, kira dan paparkan nilai *x* dan *y* atau papar mesej '*persamaan tersebut tiada penyelesaian*'.

Create a Q4Main class to test the program. It prompts the user to enter the values for a, b c, d, e, and f as well as create a LinearEquation object. Display the linear equation. Depending on the values of ad-bc, compute and display the values for x and y or display the message 'the equation has no solution'.

Contoh output:
Sample output:

```
Enter the values for a, b, c, d, e and f : 1 2 1 3 4 3<enter>

The equation :
1x + 2y = 4
1x + 3y = 3

x = 6 ; y = -1
```

(Salin fail **LinearEquation.java** dan **Q4Main.java** ke direktori akaun peperiksaan anda. Namakan semula sebagai **[matricNumberLinearEquation.java**; contoh: **WIF170001LinearEquation.java]** dan **[matricNumberQ4.java**; contoh: **WIF170001Q4.java]**).

*(Copy the **LinearEquation.java** and **Q4Main.java** files to your exam account directory. Rename as **[matricNumberLinearEquation.java**; example: **WIF170001LinearEquation.java** and **matricNumberQ4.java**; example: **WIF170001Q4.java]**).*

(4 markah/*marks*)

5. Sesuatu aktiviti yang dilengkapkan oleh pelajar akan dinilai. Sesuatu aktiviti yang dinilai boleh diberi skor angka seperti 70, 95 dan sebagainya serta gred abjad seperti A, B, C, D atau F.

An activity completed by students will be graded. A graded activity can be given a numeric score such as 70, 95 and so on, and a letter grade such as A, B, C, D or F.

a) Rekabentuk satu kelas **GradedActivity** yang terdiri daripada ahli-ahli berikut:

- Satu medan untuk skor angka suatu aktiviti yang dinilai.
- Satu kaedah *setScore* yang menetapkan skor angka.
- Satu kaedah *getScore* yang memulangkan skor angka.
- Satu kaedah *getGrade* yang memulangkan gred abjad yang sesuai dengan skor angka tersebut.

Gred	Skor angka
A	Sama dengan atau lebih besar dari 90
B	Sama dengan atau lebih besar dari 80
C	Sama dengan atau lebih besar dari 70
D	Sama dengan atau lebih besar dari 60
F	Kurang dari 60

- Satu kaedah *toString* yang memulangkan maklumat skor angka dan gred tersebut.

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

- *A field for the numeric score of a graded activity.*
- *A method setScore that sets the numeric score.*
- *A method getScore that returns the numeric score.*
- *A method getGrade that returns the letter grade that corresponds to the numeric score.*

Grade	*Numeric score*
A	*equal to or greater than 90*
B	*equal to or greater than 80*
C	*equal to or greater than 70*
D	*equal to or greater than 60*
F	*less than 60*

- *A method toString that returns information about the numeric score and grade.*

(4 markah/*marks*)

b) Rekabentuk satu kelas **Essay** yang mewarisi kelas **GradedActivity**. Kelas **Essay** tersebut perlu menentukan gred yang akan diterima oleh pelajar untuk sesuatu karangan. Jumlah markah karangan pelajar boleh mencapai sehingga 100 dan terdiri daripada empat komponen berikut:

Nahu: 30 markah (maksimum)
Ejaan: 20 markah (maksimum)
Panjang: 20 markah (maksimum)
Kandungan: 30 markah (maksimum)

Kelas ini mempunyai empat medan angka dan satu kaedah *toString* yang memulangkan maklumat mengenai empat komponen tersebut.

Design an ***Essay*** *class that inherits the* ***GradedActivity*** *class. The* ***Essay*** *class should determine the grade a student receives for an essay. The student's total essay score can be up to 100 and consists of the following four components:*

Grammar: 30 marks (maximum)
Spelling: 20 marks (maximum)
Length: 20 marks (maximum)
Content: 30 marks (maximum)

This class has four numeric fields and a toString method that returns information about the four components.

(3 markah/*marks*)

c) Bina satu kelas **Q5Main** untuk menguji program. Ia meminta pengguna memasukkan markah bagi empat komponen karangan, membina satu objek **Essay** dan memaparkan maklumat mengenai karangan tersebut.

*Create a **Q5Main** class to test the program. It asks the user to enter the marks for the four essay components, creates an **Essay** object and displays information about the essay.*

Contoh output:
Sample output:

```
Enter the marks for Grammar (maximum 30 marks): 25 <enter>
Enter the marks for Spelling (maximum 20 marks): 10 <enter>
Enter the marks for Length (maximum 20 marks): 20 <enter>
Enter the marks for Content (maximum 30 marks): 10 <enter>

Essay score:
Grammar: 25
Spelling: 10
Length: 20
Content: 10

Total score: 65
Essay grade: D
```

(Salin fail **GradedActivity.java, Essay.java dan Q5Main.java** ke akaun direktori peperiksaan anda. Namakan semula sebagai **[matricNumberGradedActivity.java , matricNumberEssay.java, matricNumberQ5.java;** contoh: **WIF170001 GradedActivity.java, WIF170001Essay.java** dan **WIF170001Q5.java**].)

(Copy the **GradedActivity.java, Essay.java** *and* **Q5Main.java** *files to your exam account directory. Rename as **[matricNumberGradedActivity.java, matricNumberEssay.java, matricNumberQ5.java**; example: **WIF170001GradedActivity.java, WIF170001Essay.java** and **WIF170001Q5.java**].)*

(3 markah/*marks*)

TAMAT
END