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 2014/2015 : SEMESTER I
ACADEMIC SESSION 2014/2015 : SEMESTER I

WXES1116 : Pengaturcaraan I
Programming I

Dis 2014/Jan 2015 Masa: 2 jam
Dec 2014/Jan 2015 Time: 2 hours

ARAHAN KEPADA CALON:
INSTRUCTIONS TO CANDIDATES:

Calon dikehendaki menjawab **SEMUA** soalan (50 markah).
*Answer **ALL** questions (50 marks).*

(Kertas soalan ini mengandungi 4 soalan dalam 6 halaman yang dicetak)
(This question paper consists of 4 questions on 6 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: **WEK100001Q1.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: **WEK100001Q1.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) {
    Random in = new Random(System.in);
    char reverse = "";
    System.out.print("Enter a word: ");
    String word = in.nextDouble();

    for (int i = length; i >= 0; i++ )
      reverse = reverse + word.nextChar(i);

    if (word == reverse)
      System.out.println("The word is a palindrome.");
    else if
      System.out.println("The word is not a palindrome.");
  }
}
```

(8 markah/*marks*)

2. Tuliskan satu aturcara untuk menentukan pemenang nombor plat BMW88. Mula-mula, aturcara tersebut meminta pengguna memasukkan bilangan tender. Kemudian, aturcara tersebut meminta nama dan harga tempahan bagi setiap tender. Selepas itu, aturcara tersebut akan memaparkan nama tender dalam urutan menurun berdasarkan harga tempahan. Akhirnya, aturcara tersebut akan memaparkan pemenang nombor plat BMW88. (Gunakan *bubblesort* untuk susunan.)

   (Simpan aturcara tersebut dalam fail **Main.java**. Salin fail ini ke direktori akaun peperiksaan anda dan namakan semula sebagai [**matricNumberQ2.java**; contoh: **WEK100001Q2.java**].

   Write a program to decide the winner of the BMW88 number plate. First, the program requests the user to enter the number of tender. Then, the program requests the name and reservation price from each tender. After that, the program displays the name of the tender in decreasing order of their reservation price. Finally, the program should display the winner of the BMW88 number plate. (Use bubblesort for sorting.)

   *(Save the program in the **Main.java** file. Copy this file to your exam account directory and rename as [**matricNumberQ2.java**; example: **WEK100001Q2.java**].*

   Contoh output:
   Sample output:

```
Enter number of tender : 5
Enter name : Rizal
Enter reservation price : 3400.20
Enter name : John Lim
Enter reservation price : 2850.50
Enter name : Rafiq
Enter reservation price : 9860.00
Enter name : Mutusamy
Enter reservation price : 6835.20
Enter name : Kent Liew
Enter reservation price : 4888.00
List of tender :
Rafiq, Mutusamy, Kent Liew, Rizal, John Lim
The winner of BMW88 : Rafiq (9860.00)
```

(12 markah/*marks*)

3. Fail **Q3.txt** mengandungi jumlah pesanan untuk produk D, E dan F. Tuliskan satu aturcara untuk mengira jumlah bilangan pesanan untuk setiap produk. Kemudian, paparkan jumlah pesanan bagi setiap jenis produk dan jenis produk yang mempunyai jumlah pesanan yang terbanyak. Simpankan output tersebut ke dalam fail yang bernama **summary.txt**.

   (Salin fail **Q3.txt** dari direktori akaun peperiksaan anda. Simpan aturcara dalam fail **Main.java**. Salin fail **Main.java** ini ke akaun direktori peperiksaan anda and namakan ia semula sebagai [**matricNumberQ3.java**; contoh: **WEK100001Q3.java**].

   *The **Q3.txt** file contains the total number of orders for product D, E and F. Write a program that computes the total number of orders for each product. Then, display the total number of orders for each product and the type of product with the highest number of order. Save the output to a file named **summary.txt**.*

   *(Copy the **Q3.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 it as **[matricNumberQ3.java**; example: **WEK100001Q3.java]**.*

   Contoh output:
   Sample output:

```
Total number of orders [Product D]: 1742
Total number of orders [Product E]: 2641
Total number of orders [Product F]: 613
Product E has the highest number of order.
```

(12 markah/*marks*)

4. Reka bentuk satu kelas **Hero** yang terdiri daripada ahli berikut:
   - Satu medan untuk nama pahlawan.
   - Satu medan untuk jenis pahlawan.
   - Satu medan untuk mata hayat pahlawan.
   - Satu pembina yang mengandungi nama dan jenis pahlawan.
   - Kaedah pencapai dan *mutator* untuk nama dan mata hayat pahlawan.
   - Satu kaedah *display* yang memaparkan nama, jenis dan mata hayat pahlawan.

   *Design a **Hero** class that consists of the following members:*
   - *A field for the name of the hero.*
   - *A field for the type of the hero.*
   - *A field for the health points of the hero.*
   - *A constructor that contains the name and type of the hero.*
   - *Accessor and mutator method for name and health points of the hero.*
   - *A display method to display the hero's name, type and health points.*

(5 markah/*marks*)

Reka bentuk satu kelas **Strength** yang mewarisi kelas **Hero**. Kelas **Strength** sepatutnya mempunyai ahli berikut:

- Satu pembina.
- Kaedah *damage* yang memulangkan nilai kerosakan kepada lawan dan memaparkan nilai kerosakan tersebut. Pahlawan *strength* menyebabkan kerosakan rawak di antara 3 hingga 10.

*Design a **Strength** class that extends the **Hero** class. The **Strength** class should have the following members:*

- *A constructor*
- *A damage method that returns the damage to the opponent and displays the damage. The strength hero causes a random damage in between 3 to 10 inclusive.*

(4 markah/*marks*)

Reka bentuk satu kelas **Agility** yang mewarisi kelas **Hero**. Kelas **Agility** sepatutnya mempunyai ahli berikut:

- Satu pembina.
- Kaedah *damage* yang memulangkan nilai kerosakan kepada lawan dan memaparkan nilai kerosakan tersebut. Kerosakan dari pahlawan *agility* adalah dikira berdasarkan formula di bawah:

  *damage = agility damage * special skill*

  (*agility damage* ialah kerosakan rawak di antara 1 hingga 5 dan *special skill* ialah kerosakan rawak di antara 0 hingga 3.)

*Design an **Agility** class that extends the **Hero class**. The **Agility** class should have the following members:*

- *A constructor.*
- *A damage method that returns the damage to the opponent and displays the damage. The damage for agility hero is calculated based on the formula below:*

  *damage = agility damage * special skill*

  (The agility damage is a random damage in between 1 to 5 inclusive and the special skill is a random damage in between 0 to 3 inclusive.)

(4 markah/*marks*)

Bina satu kelas **Main** menggunakan kod di bawah untuk menguji aturcara tersebut. Bina satu kaedah **playGame** yang memilih seorang pahlawan secara rawak untuk memulakan permainan. Permainan akan berterusan sehingga mata hayat salah satu pahlawan mencapai sifar atau kurang daripada sifar. Kemudian, kaedah tersebut akan memaparkan pemenang permainan.

*Create a **Main** class using the code below to test the program. Create a method **playGame** that randomly selects one of the heroes to start the game. The game will continue until the health points of one of the hero reaches zero or less than zero. The method will then display the winner of the game.*

```
public static void main(String[] args) {
   Strength a = new Strength("Chaos Knight", "Strength", 30);
   Agility b = new Agility("Bloodseeker", "Agility", 25);
   a.display();
   b.display();
   System.out.println();
   playGame(a,b);
}
```

(Salin fail **Hero.java, Strength.java, Agility.java** dan **Main.java** ke dalam direktori akaun peperiksaan anda.)

*(Copy the **Hero.java, Strength.java, Agility.java** and **Main.java** files to your exam account directory.)*

Contoh output:
Sample output:

```
Chaos Knight (Strength) - HP: 30
Bloodseeker (Agility) - HP: 25

Bloodseeker damage 12 *** Chaos Knight (Strength) - HP: 18

Chaos Knight damage 5 *** Bloodseeker (Agility) - HP: 20

Bloodseeker damage 3 *** Chaos Knight (Strength) - HP: 15

Chaos Knight damage 7 *** Bloodseeker (Agility) - HP: 13

Bloodseeker damage 8 *** Chaos Knight (Strength) - HP: 7

Chaos Knight damage 7 *** Bloodseeker (Agility) - HP: 6

Bloodseeker damage 10 *** Chaos Knight (Strength) - HP: -3

Bloodseeker wins the game!
```

(5 markah/*marks*)

TAMAT
END