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 2018/2019 : SEMESTER II
ACADEMIC SESSION 2018/2019 : SEMESTER II

WIX1002 : Asas-Asas Pengaturcaraan
Fundamentals of Programming

Jun 2019
June 2019

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 6 soalan dalam 9 halaman yang dicetak)
(This question paper consists of 6 questions on 9 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.)*

```
// Filename: Q1.java
//The purpose of this program is to count and display the total number of even integers in the array.
//Another purpose is to find the biggest integer in  the array and display it.
//The output based on the given list, should be as below.
//The number of even integers is 3.
//The biggest integer is 74

import java.util.Scanner ;
public class Q1 {
  public static void main(String[] args) {
    int[] num = {"66", "15", "20", "27", "74", "33"};
    int cnt=0;
    for (int k = 0; k < num.length(); k++ )
      if (isEven(num))
          cnt++;

    System.out.println("The number of even integers is " + cnt);
    System.out.println("The biggest integer is " + findMax(num[0]));
  }

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

 public static void findMax(int[] a) {
   int max = a[0];
   for (int n = 0; n < a.length; n++ )
     if (a[n] > max)
         max = a[n] ;
   return max;
 }

} // end class
```

(5 markah/*marks*)

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

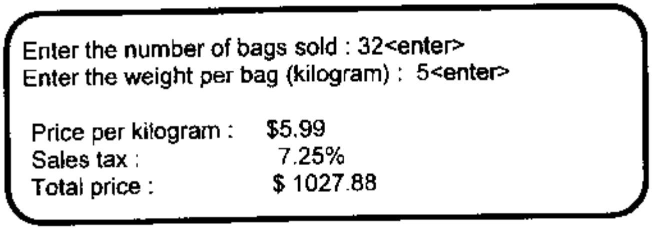
*totalPrice = unitWeight * numberOfBags * 5.99 ;*
*totalPriceWithTax = totalPrice + totalPrice * 0.0725 ;*

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, 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 :            $ 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. Tulis satu program yang membenarkan pelajar darjah satu berlatih penambahan. Program akan menjana secara rawak dua integer-satu-digit, **number1** dan **number2**, dan paparkan kepada pelajar satu soalan seperti "Apakah 7 + 9 ?", seperti ditunjuk dalam contoh output di bawah. 7 dan 9 adalah nombor rawak yang dijana untuk **number1** dan **number2**. Selepas pelajar taip jawapan, program akan paparkan satu mesej menyatakan benar atau palsu. Kemudian program akan tanya pelajar jika dia ingin mencuba soalan lain.

*Write a program to let a first-grader practice addition. The program randomly generates two single-digit integers, **number1** and **number2**, and displays to the student a question such as "What is 7 + 9 ?", as shown in the sample output below. 7 and 9 are the random numbers generated for **number1** and **number2**. After the student types the answer, the program* *displays a message to indicate whether it is true or false. Then, the program will ask the student if he wants to try another question.*

Contoh output:
Sample output:

```
What is 7 + 9 ? 16 <enter>
7 + 9 = 16 is true
Do you want to try another question (y/n) ? : n
Program ends.
```

```
What is 1 + 7 ? 9 <enter>
1 + 7 = 9 is false
Do you want to try another question (y/n) ? : y

What is 6 + 6 ? 12 <enter>
6 + 6 = 12 is true
Do you want to try another question (y/n) ? : y

What is 0 + 3 ? 3 <enter>
0 + 3 = 3 is true
Do you want to try another question (y/n) ? : n
Program ends
```

(8 markah/*marks*)

(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]**.)*

4. a) Takrifkan satu kelas bernama ***Pizza*** yang mengandungi medan dan kaedah seperti disenaraikan dalam rajah UML berikut:

Define a class named Pizza that contains the fields and methods as listed in the following UML diagram:

Pizza	
-size : String	This holds the size of the pizza whether large, medium or small
-numOfCheeseTopping : int	This holds the number of cheese toppings
-numOfBeefTopping : int	This holds the number of beef toppings
-numOfChickenTopping : int	This holds the number of chicken toppings

+Pizza () +Pizza (size:String, numOfCheeseTopping : int numOfBeefTopping : int numOfChickenTopping : int)	The constructor with the data fields
+getSize() : String +getNumOfCheeseTopping() : int +getNumOfBeefTopping () : int +getNumOfChickenTopping () : int	Accessors to get the four values
+setSize() : String +setNumOfCheeseTopping () : int +setNumOfBeefTopping () : int +setNumOfChickenTopping () : int	Mutators to set the four values.
+computeCost():double	The *computeCost* method calculates the cost of the pizza : Small : $10 + $2 per topping Medium : $12 + $2 per topping Large : $14 + $2 per topping
+display() : void	A description of the pizza object to be displayed consisting of the pizza size, quantity of each topping and cost.

(6 markah/*marks*)

b) Tulis satu program ujian, ***Q4Main*** yang membina satu objek *Pizza* dengan data yang bersesuaian. Paparkan objek tersebut. Satu contoh output diberikan di bawah untuk satu *pizza* besar dengan satu *topping* keju, daging dan ayam.

Write a tester program, ***Q4Main*** *that creates a Pizza object with appropriate data. Display the object. A sample output is given below for a large pizza with one cheese, beef and chicken topping.*

(2 markah/*marks*)

(Simpan aturcara tersebut dalam fail **Pizza.java** dan **Q4Main.java**. Salin fail ini ke direktori akaun peperiksaan anda dan namakan semula sebagai [**matricNumberQ4.java**; contoh: **WIF170001Q4.java**] dan [**matricNumberPizza.java**; contoh : **WIF170001Pizza.java**])

(Save the program in the **Pizza.java** *and* ***Q4Main.java*** *file. Copy this file to your exam account directory and rename as [**matricNumberQ4.java**; example:* ***WIF170001Q4.java**] and [**matricNumberPizza**.java; example :* ***WIF170001Pizza.java]*** *)*

5. a) Takrif satu kelas bernama ***quadraticEquation*** untuk satu sistem persamaan kuadratik $ax^2 + bx + c = 0$. Kelas tersebut mengandungi:

- *medan data persendirian a,b dan c yang mewakili tiga pekali.*
- *satu pembina dengan hujah untuk a, b dan c.*
- *tiga kaedah **get** untuk a, b dan c.*
- *satu kaedah bernama **Discriminant()** yang pulangkan diskriminan b^2-4ac.*
- *Kaedah-kaedah bernama **calcRoot1()** dan **calcRoot2()** yang pulangkan **r1** dan **r2**, dua akar persamaan tersebut. [Pembayang : guna **Math.sqrt**$(\boldsymbol{n})$ untuk cari punca kuasa dua n]*

$$r1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad r2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Kaedah **calcRoot1** dan **calcRoot2** tersebut adalah berguna hanya jika **Discriminant** adalah positif. Kedua kaedah akan memulangkan 0 jika **Discriminant** adalah negatif.

*Define a class named **QuadraticEquation** for a quadratic equation* $ax^2 + bx + c = 0$.*The class contains:*

- *private data fields a, b, and c that represent three coefficients.*
- *a constructor with the arguments for a, b and c.*
- *three get methods for a, b and c.*
- *a method named **Discriminant()** that returns the discriminant which is b^2-4ac.*
- *methods named **calcRoot1()** and **calcRoot2()** that return **r1** and **r2**, the two roots of the equation. [Hint : use **Math.sqrt(n)** to find the square root of n]*

$$r1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad r2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

*The methods **calcRoot1** and **calcRoot2** are useful only if **Discriminant** is positive. Both methods will return 0 if **Discriminant** is negative.*

(6 markah/*marks*)

b) Tulis satu program ujian yang membina objek *quadraticEquation* dan meminta pengguna memasukkan nilai untuk a, b, dan c serta paparkan jawapan berdasarkan ***Discriminant***. Jika ***Discriminant*** adalah positif, paparkan nilai r1 dan r2. Jika ***Discriminant*** adalah 0, paparkan akar tersebut. Selainnya, paparkan mesej "*the equation has no roots*".

Write a tester program that creates the quadraticEquation object and prompts the user to enter values for a, b and c; and displays the answer based on **Discriminant**. *If* **Discriminant** *is positive, display the value of r1 and r2. If* **Discriminant** *is 0, display the root. Otherwise, display the message "the equation has no roots."*

Contoh output:
Sample output:

```
Enter values for a, b and c : 1 1 1 <enter>
The equation is : 1x(^2) +(1) x + (1)
Discriminant : -3
The equation has no roots
```

```
Enter values for a, b and c : 1 0 -1 <enter>
The equation is : 1x(^2) + (0)x + (-1)
Discriminant : 4
The roots : -1 and 1
```

```
Enter values for a, b and c : 1 2 1 <enter>
The equation is : 1x(^2) + (2)x + (1)
Discriminant : 0
Same roots : -1
```

(4 markah/*marks*)

(Simpan aturcara tersebut dalam fail **QuadraticEq.java** dan **Q5Main.java**. Salin fail ini ke direktori akaun peperiksaan anda dan namakan semula sebagai [**matricNumberQuadraticEq.java**; contoh: **WIF170001QuadraticEq.java**] dan [**matricNumberQ5.java**; contoh: **WIF170001Q5.java**].)

(Save the program in the **QuadraticEq.java** and ***Q5Main.java*** *file. Copy these files to your exam account directory and rename as* ***[matricNumber*QuadraticEq.java*.java*;** *example:* ***WIF170001*QuadraticEq.java*.java]*** *and* ***[matricNumberQ5.java*;** *example:* ***WIF170001Q5.java]**.)*

6. a) Takrifkan satu kelas bernama *Order* yang melakukan pemprosesan penempahan sesuatu produk dan sub kelasnya *ShippedOrder* yang mengira harga barang termasuk kos penghantaran. Kelas *Order* mengandungi:

- dua medan data *String* bernama ***customer name*** dan ***customer ID.***
- satu medan data *int* bernama ***quantity ordered***.
- satu medan data *double* bernama ***unit price***.
- satu pembina tanpa hujah dengan nilai lalai yang bersesuaian.
- satu pembina yang mencipta ***Order*** dengan nilai yang dinyatakan.
- Kaedah pencapai dan pemutat untuk medan data.
- satu kaedah ***computeTotalPrice()*** yang pulangkan:
  *totalPrice = quantity * unit price*
- satu kaedah **toString()** yang pulangkan satu deskripsi rentetan untuk ***Order***.

Define a class named Order that performs the order processing of a product and its subclass, ShippedOrder that calculates the product price including shipping costs. The Order class contains:

- *two String data fields named **customer name** and **customer ID.***
- *one int data field named **quantity ordered**.*
- *one double data field named **unit price**.*
- *a no-argument constructor with appropriate default values.*
- *a constructor that creates **Order** with the specified data values.*
- *accessor and mutator methods for the data fields.*
- *a method **computeTotalPrice()** that returns:*
  *totalPrice = quantity * unit price*
- *a method **toString()** that returns a string description for the **Order**.*

(8 markah/*marks*)

b) Takrifkan sub kelas *ShippedOrder*. Ia mengandungi:

- satu kaedah **computeTotalPrice()** yang mengatasi **computeTotalPrice** dengan menambah satu caj penghantaran dan pengendalian sebanyak $4.00.
- satu kaedah **toString()** yang pulangkan deskripsi rentetan untuk **ShippedOrder.**

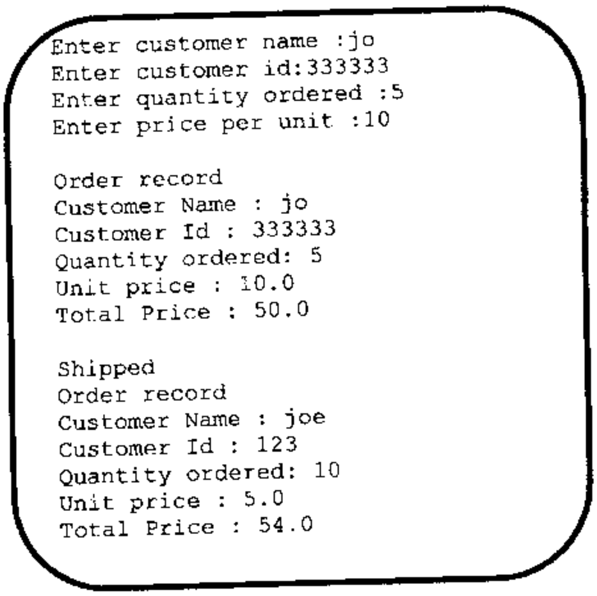
Define the ShippedOrder subclass. It contains:

- *a method **computeTotalPrice()** that overrides computeTotalPrice by adding a shipping and handling charge of $4.00.*
- *a method **toString()** that returns a string description for the ShippedOrder.*

(4 markah/*marks*)

c) Tulis satu program ujian yang membina satu objek *Order* di mana data diberikan oleh pengguna. Paparkan keputusannya. Begitu juga, bina satu objek *ShippedOrder* dengan data yang dibekal pengguna untuk objek tersebut dan paparkan keputusannya. Contoh output diberi di bawah.

Write a test program that creates one Order object with its data provided by the user Display the results. Similarly, create one ShippedOrder object with user provided data and display the results. A sample output is given below.



```
Enter customer name :jo
Enter customer id:333333
Enter quantity ordered :5
Enter price per unit :10

Order record
Customer Name : jo
Customer Id : 333333
Quantity ordered: 5
Unit price : 10.0
Total Price : 50.0

Shipped
Order record
Customer Name : joe
Customer Id : 123
Quantity ordered: 10
Unit price : 5.0
Total Price : 54.0
```

(2 markah/*marks*)

(Simpan aturcara tersebut dalam fail **Order.java**, **ShippedOrder.java** dan **Q6Main.java**. Salin fail ini ke direktori akaun peperiksaan anda dan namakan semula sebagai [**matricNumberOrder.java**; contoh: **WIF170001Order.java**]; **matricNumberShippedOrder.java**; contoh: **WIF170001ShippedOrder.java**] dan **matricNumberQ6.java**; contoh: **WIF170001Q6.java**])

*(Save the program in the **Order.java, ShippedOrder.java** and **Q6Main.java** file. Copy these files to your exam account directory and rename as [**matricNumberOrder.java**; example: **WIF170001Order.java**; **matricNumberShippedOrder.java**; example: **WIF170001ShippedOrder.java**] and **matricNumberQ6.java**; example: **WIF170001Q6.java**])*

TAMAT
END