

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 2012/2013 : SEMESTER II
ACADEMIC SESSION 2012/2013 : SEMESTER II

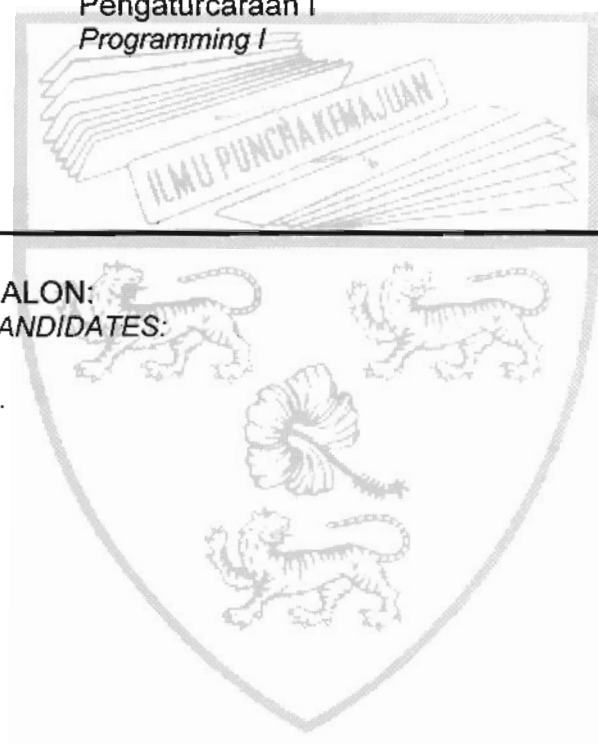
WXES1116 : Pengaturcaraan I
Programming I

Jun 2013
June 2013

Masa: 2 jam
Time: 2 hours

ARAHAN KEPADA CALON:
INSTRUCTIONS TO CANDIDATES:

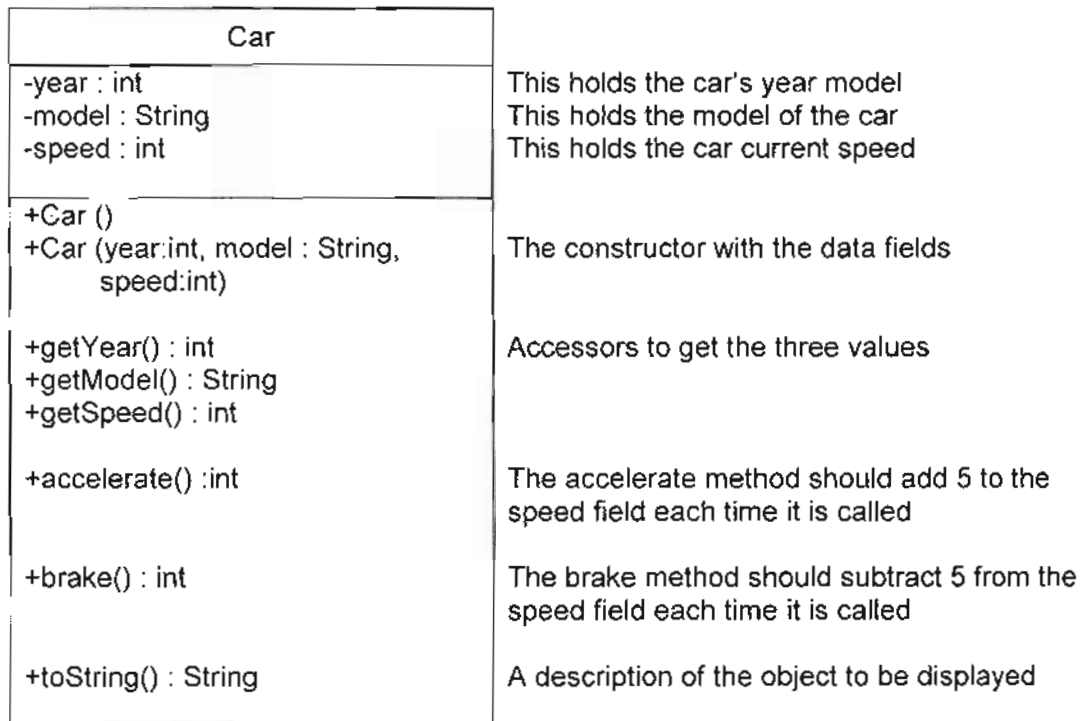
Jawab semua soalan.
Answer all questions.



(Kertas soalan ini mengandungi 4 soalan dalam 6 halaman yang dicetak)
(This question paper consists of 4 questions on 6 printed pages)

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

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



(6 markah/marks)

- b) Tulis satu program, *TestCar* yang membina satu objek *Car* dengan data yang bersesuaian. Paparkan objek tersebut. Panggil kaedah *accelerate* sebanyak lima kali. Selepas setiap panggilan ke kaedah *accelerate*, dapatkan kelajuan semasa kereta tersebut dan paparkannya. Kemudian panggil kaedah *brake* sebanyak lima kali. Selepas setiap panggilan ke kaedah *brake*, dapatkan kelajuan semasa kereta tersebut dan paparkannya. Satu contoh output diberikan di bawah:

Write a program, TestCar that creates a Car object with appropriate data. Display the object. Call the accelerate method five times. After each call to the accelerate method, get the current speed of the car and display it. Then, call the brake method five times. After each call to the brake method, get the current speed of the car and display it. A sample output is given below:

```
Car Model : Nissan Xtrail  
Car Year : 2006  
Initial Speed : 80
```

```
Car speed now : 85  
:  
Car speed now : 100  
Car speed now : 95  
Car speed now : 80  
:  
Car speed now : 75
```

(6 markah/marks)

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

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 the two roots of the equation.

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

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

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

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 test 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."

(6 markah/marks)

3. 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:

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 number.*
- *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.*

(4 markah/marks)

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

Define the ShippedOrder class. 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) Lukis gambarajah UML untuk kedua-dua kelas.

Draw UML diagrams for both classes.

(4 markah/marks)

- d) 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.

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.

(4 markah/marks)

4. a) Takrifkan satu kelas bernama *Location* untuk menyimpan nilai terkecil dan lokasinya dalam satu tatasusunan 1 dimensi. Kelas ini mengandungi:

Define a class named Location for storing the smallest value and its location in a one-dimensional array. The class contains:

- two public data fields named **whichIndex** and **minValue** where **whichIndex** stores the index of the smallest element in the array and **minValue** stores the smallest element in the array.
- a no-argument constructor with default values as 0 and 0
- a constructor that creates a *Location* with the specified list of positive numbers.
- a method **locateMin()** that returns an **instance** of *Location*:
`public static Location locateMin(int[] num)`

(6 markah/marks)

- b) Tulis satu program ujian yang membina satu objek *Location* dengan tatasusunan {100,95,64,66} dan paparkan lokasi nombor terkecil dalam tatasusunan dan nombor terkecil tersebut seperti berikut:

Write a test program that creates a Location object with array {100,95,64,66} and displays the location of the smallest number in that array and the number as follows:

The location of the smallest element is at index 2 and the element is 64.

(4 markah/marks)

**TAMAT
END**