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

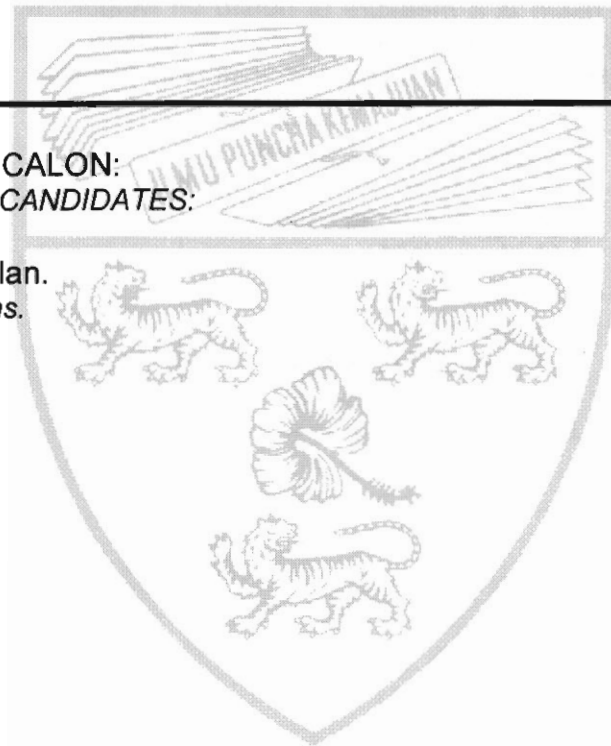SESI AKADEMIK 2011/2012 : SEMESTER II
ACADEMIC SESSION 2011/2012 : *SEMESTER II*

WXES1116 : Pengaturcaraan I
Programming I

Jun 2012   Masa : 2 jam
June 2012   *Time : 2 hours*

ARAHAN KEPADA CALON:
INSTRUCTIONS TO CANDIDATES:

Jawab **SEMUA** soalan.
*Answer **ALL** questions.*

(Kertas soalan ini mengandungi 5 soalan dalam 6 halaman yang dicetak)
(This question paper consists of 5 questions on 6 printed pages)

1. Terangkan dengan ringkas konsep *OOP* berikut:

   Explain briefly the following OOP concepts:

   a) *class*
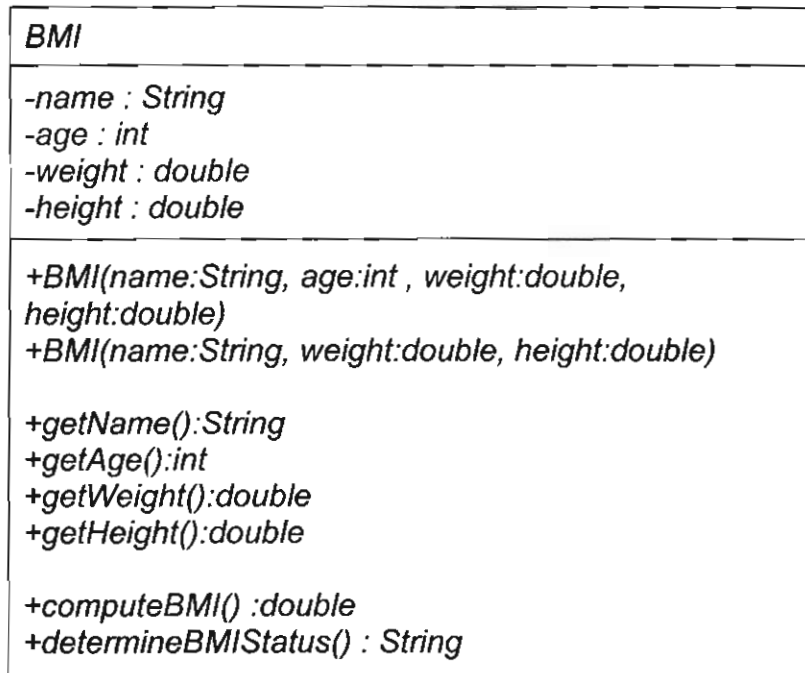
   b) *inheritance*

   (3 markah/*marks*)

2. Indeks jisim badan (*BMI*) adalah satu ukuran kesihatan ke atas berat. Ia boleh dikira dengan membahagikan berat dalam kilogram dengan gandadua ketinggian dalam unit meter. Interpretasi *BMI* untuk mereka yang berumur 16 tahun ke atas adalah seperti berikut:

   Body mass index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing it by the square of your height in meters. The interpretation of BMI for people 16 years or older is as follows:

BMI	Status (interpretation)
Below 16	Seriously underweight
16-18	Underweight
18-24	Normal weight
24-29	Overweight
29-35	Seriously overweight
Above 35	Gravely overweight

a) Takrifkan kelas *BMI* berdasarkan gambarajah *UML* di bawah.

Define the BMI class based on the UML diagram below.

```
BMI
------------------------------------------------
-name : String
-age : int
-weight : double
-height : double
------------------------------------------------
+BMI(name:String, age:int , weight:double,
height:double)
+BMI(name:String, weight:double, height:double)

+getName():String
+getAge():int
+getWeight():double
+getHeight():double

+computeBMI() :double
+determineBMIStatus() : String
```

(6 markah/*marks*)

b) Tulis satu program *TestBMI* yang membina satu objek *BMI* di mana nama = "Mr. Bean", umur = 18, berat = 80 kilogram dan ketinggian = 1.6 meter dan paparkan maklumat seperti contoh di bawah.

Write a test program, TestBMI that creates a BMI object where name = "Mr. Bean", age = 18, weight = 80 kilogram and height = 1.6 meters and display the information as illustrated below.

> The BMI for Mr. Bean is 31.25 seriously overweight

(4 markah/*marks*)

3. 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 equation:

$$ax + by = e \quad \text{where} \quad x = \frac{ed-bf}{ad-bc} \qquad y = \frac{af-ec}{ad-bc}$$
$$cx + dy = f$$

   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.
   - six get methods for a, b, c, d, e and f.
   - a method named **isSolvable()** that returns true if ad-bc is not 0.
   - methods **computeX()** and **computeY()** that return the solution for the equation.

   (6 markah/*marks*)

   b) Lukis gambarajah *UML* untuk kelas tersebut.

   Draw the UML diagram for the class.

   (3 markah/*marks*)

   c) Tulis satu program ujian yang meminta pengguna memasukkan nilai untuk a, b, c, d, e dan f serta paparkan jawapan untuk x dan y dengan melaksanakan kelas tersebut. Jika ad-bc adalah 0, paparkan mesej *"the equation has no solution"*.

   Write a test program that prompts the user to enter values for a, b, c, d, e and f and displays the answer for x and y by implementing the class. If ad-bc is 0, display the message "the equation has no solution."

   (4 markah/*marks*)

4. a) Takrifkan satu kelas bernama *Circle* dan sub kelasnya *Cylinder*. Kelas *Circle* mengandungi:

   Define a class named Circle and its subclass Cylinder. The Circle class contains:

   - one double data field named radius.
   - a no-argument constructor with default values as 0.0.
   - a constructor that creates a Circle with the specified radius.
   - accessor method for the data field.
   - a method **area()** that returns the area of the circle:
     area = π (radius^2)
   - a method **perimeter()** that returns the perimeter of the circle:
     perimeter = 2 (π) (radius)
   - a method **toString()** that returns a string description for the circle.

   (3 markah/*marks*)

   b) Takrifkan kelas *Cylinder* tersebut. Ia mengandungi:

   Define the Cylinder class. It contains:

   - one double data field named height.
   - a no-argument constructor with default values as 0.0.
   - a constructor that creates a Cylinder with the specified radius and height.
   - accessor method for the data field.
   - a method **area()** that returns the surface area of the Cylinder:
     area = 2 (π) (radius^2) + 2 (π) (radius) height
   - a method **volume()** that returns the volume of the Cylinder:
     volume = π (radius^2) height
   - a method **toString()** that returns a string description for the Cylinder.

   (5 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 *Circle* dengan radius = 15 dan satu objek *Cylinder* dengan radius = 20 dan ketinggian = 30 serta paparkan deskripsi berkenaan objek-objek tersebut dengan melaksanakan kelas tersebut.

   Write a test program that creates a Circle object with radius = 15 and a Cylinder object with radius = 20 and height = 30 and displays the description of the objects by implementing the class.

   (2 markah/*marks*)

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

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

   - two public data fields named **row** and **maxValue** where **row** stores the index of the biggest element in the array and **maxValue** stores the biggest 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 numbers.
   - a method **locateMax()** that returns an **instance** of Location:
        public static Location **locateMax**(int[] num)

   (6 markah/*marks*)

   b) Tulis satu program ujian yang membina satu objek *Location* dengan tatasusunan {4,99,2} dan paparkan lokasi nombor terbesar dalam tatasusunan tersebut seperti berikut:

   Write a test program that creates a Location object with array {4,99,2} and displays the location of the biggest number in that array as follows:

   > *The location of the biggest element is at index[1] and the element is 99.*

   (4 markah/*marks*)

   TAMAT
   END