

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 2016/2017 : SEMESTER I
ACADEMIC SESSION 2016/2017 : SEMESTER I

WIX1002 : Asas-Asas Pengaturcaraan
Fundamentals of Programming

Dis 2016/Jan 2017
Dec 2016/Jan 2017

Masa: 3 jam 30 minit
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 9 halaman yang dicetak)
(This question paper consists of 5 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: **WIF160001Q1.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: **WIF160001Q1.java**] and copy this file to your exam account directory.)*

```
import java.util.Scanner;
//Filename:Q1.java
public class GraphsTest
{ public static void main( String args[] )
{ Scanner input = new Scanner( System.in );
  int number1 = 0, number2 = 0, number3 = 0, number4 = 0, number5 = 0;
  int inputNumber; // number entered by user
  int value = 0; // number of stars to print
  counter = 1; // counter for current number
  while ( int counter <= 5 )
  { System.out.print( "Enter number: " );
    inputNumber = input.nextInt();
    // define appropriate num if input is between 1-30
    if ( inputNumber >= 1 && inputNumber <= 30 )
    { switch ( inputNumber )
      { case 1: number1 = inputNumber; break;
        case 2: number2 = inputNumber; break;
        case 3: number3 = inputNumber;
          4: number4 = inputNumber; break;
        case 5: number5 = inputNumber; break;
        counter++
      } // end if
    } else
      System.out.println( "Invalid Input\nNumber should be between 1 and 30" );
    } // end while
  // print histograms
  for ( counter = 1, counter >= 5, counter++ )
  { switch ( counter == 1 )
    { case 1: value = number1; break;
      case 2: value = number2; break;
      case 3: value = number3; break;
      case 4: value = number4; break;
      case 5: value = number5; break;
    } // switch
    for ( int j = 1; int j <= value; int j++ )
      System.out.print( "*" );
    System.out.println();
  } // end for loop
} // end main
} // end class Graphs
```

(10 markah/marks)

2. (Aplikasi kewangan: Mengira yuran pengajian masa hadapan) Tulis satu aturcara yang mengandungi satu kaedah, *computeFee*, untuk mengira yuran pengajian bagi satu tahun tertentu. Kaedah tersebut menerima kedua-dua yuran pengajian permulaan dan kadar peningkatan setiap tahun. Kaedah tersebut juga menerima tahun di mana yuran pengajian tersebut akan ditentukan. Gunakan pengulangan di dalam kod anda. Dapatkan nilai yuran pengajian permulaan, kadar peningkatan yuran dan juga tahun di dalam kaedah utama sebelum menghantarnya sebagai hujah kepada *computeFee*. Kaedah ini akan memulangkan yuran pengajian yang dikira kepada kaedah utama.

(Financial application: Computing future tuition fee). Write a program containing a method, computeFee, to compute the tuition fee for a particular year. The method accepts both the initial tuition fee, and the rate of increment every year. The method also accepts the year in which a particular tuition fee is to be determined. Use repetition in your code. Get the values of the initial tuition fee, rate of increment and the year within your main method before passing it as arguments to computeFee. This method will return the computed tuition fee to the main method.

Contoh output:

Sample output:

```
Enter the initial tuition fee (i.e year 1): 100
Enter the yearly rate of increment (e.g enter 5.2 for 5.2%): 5.2
Enter the year for which you wish to compute the tuition fee for: 3

Computed tuition fee for year 3 is: 110.25
```

(Simpan aturcara tersebut dalam fail **Q2Main.java**. Salin fail ini ke direktori akaun peperiksaan anda dan namakan semula sebagai [**matricNumberQ2.java**; contoh: **WIF160001Q2.java**].)

*(Save the program in the **Q2Main.java** file. Copy this file to your exam account directory and rename as [**matricNumberQ2.java**; example: **WIF160001Q2.java**].)*

(10 markah/marks)

3. Tulis satu aturcara yang menjana satu rentetan enam aksara secara rawak daripada Jadual ASCII dan isihnya dalam turutan menaik dan menurun. Rentetan tersebut akan dijana daripada nombor rawak yang dijana berdasarkan jadual ASCII. Rentetan yang dijana itu perlu disimpan di dalam satu fail teks bernama **data.txt**. Seterusnya, aturcara ini akan mengisih rentetan tersebut dalam turutan menaik dan menurun serta memaparkan hasil kedua-dua rentetan yang telah diisih tersebut. Rentetan yang asal juga perlu dipaparkan pada akhir aturcara dengan membaca fail teks **data.txt** yang telah dibina pada awal aturcara. Aturcara in hendaklah merangkumi:

- Satu rentetan (6 huruf) berdasarkan turutan rawak dari Jadual ASCII. Nombor rawak tersebut mestilah dalam lingkungan nombor dari jadual di bawah. Hanya abjad dibenarkan.
- Satu fail teks untuk menyimpan dan membaca rentetan yang telah dijana.
- Satu kaedah isihan yang menerima dua hujah. Hujah pertama adalah tatasusunan aksara yang perlu diisih. Hujah kedua adalah satu simbol untuk menandakan samada isihan tersebut dalam mod turutan menaik '>' atau menurun '<'. Mekanisma isihan yang digunakan adalah *Bubble Sort*.

*Write a program that generates a random six-character string from the ASCII table and sort it in ascending and descending order. The string will be generated from the random numbers generated based on the ASCII table. The generated string has to be stored in a text file named **data.txt**. The program will then sort the string in ascending and descending order and display the output for both the sorted strings. The original string should also be displayed at the end of the program by reading the text file, **data.txt**, created earlier. The program should include:*

- *A string (6 letters) based on the random sequence from the ASCII table. The random number must in the range of values from the table below. Only alphabets are allowed.*
- *A text file to store and read the generated string.*
- *A sort method that accepts two arguments. The first argument is a character array that needs to be sorted. The second argument is a symbol to denote whether the sorting is ascending '>' or descending '<' mode. The sorting mechanism used is Bubble Sort.*

ASCII TABLE

65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Jadual ASCII:
ASCII table:

Contoh output:

Sample output:

The string generated is: SsJeFu
 String sorted in ascending order: FJSesu
 String sorted in descending order: useSJF
 Original string from file: SsJeFu

(Simpan aturcara tersebut dalam fail **Q3Main.java**. Salin fail ini ke direktori akaun peperiksaan anda dan namakan semula sebagai [**matricNumberQ3.java**; contoh: **WIF160001Q3.java**]. Fail **data.txt** juga perlu disalin ke direktori akaun peperiksaan anda.)

*(Save the program in the **Q3Main.java** file. Copy this file to your exam account directory and rename as [**matricNumberQ3.java**; example: **WIF160001Q3.java**]. The file, **data.txt** should also be copied to your account directory.)*

(10 markah/marks)

4. Nombor kompleks adalah satu nombor yang boleh diungkapkan dalam bentuk $a + bi$ di mana a dan b adalah nombor nyata dan i adalah unit khayalan yang memenuhi persamaan $i^2 = -1$. Dalam ungkapan ini, a adalah bahagian nyata dan b adalah bahagian khayalan nombor kompleks tersebut.

A complex number is a number that can be expressed in the form $a + bi$, where a and b are real numbers and i is the imaginary unit which satisfies the equation $i^2 = -1$. In this expression, a is the real part and b is the imaginary part of the complex number.

- a) Rekabentuk kelas **Complex** untuk melakukan pengiraan dengan nombor kompleks. Kelas ini terdiri daripada ahli-ahli berikut:

- Satu medan *double* untuk bahagian nyata.
- Satu medan *double* untuk bahagian khayalan.
- Satu pembina tanpa hujah.
- Satu pembina dengan hujah.
- Satu kaedah *addComplexNum* untuk menjumlah dua nombor kompleks menggunakan formula ini: bahagian nyata dijumlahkan bersama dan bahagian khayalan dijumlahkan bersama.

$$(a + bi) + (c + di) = (a+c) + (b+d)i$$

- Satu kaedah *subtractComplexNum* untuk penolakan dua nombor kompleks menggunakan formula ini: bahagian nyata operan kanan ditolak dari bahagian nyata operan kiri dan bahagian khayalan operan kanan ditolak dari bahagian khayalan operan kiri.

$$(a + bi) - (c + di) = (a-c) + (b-d)i$$

- Satu kaedah *toString* yang akan paparkan satu nombor kompleks dalam bentuk (a, b) di mana a adalah bahagian nyata dan b adalah bahagian khayalan.

Design a **Complex** class to perform computation with complex numbers. This class consists of the following members:

- A double field for the real part.
- A double field for the imaginary part.
- A no-argument constructor.
- A constructor with arguments.
- A method *addComplexNum* to add two complex numbers using this formula: the real parts are added together and the imaginary parts are added together.

$$(a + bi) + (c + di) = (a+c) + (b+d)i$$

- A method *subtractComplexNum* to subtract two complex numbers using this formula: the real part of the right operand is subtracted from the real part of the left operand and the imaginary part of the right operand is subtracted from the imaginary part of the left operand.

$$(a + bi) - (c + di) = (a-c) + (b-d)i$$

- A method *toString* that will display a complex number in the form (a, b) where a is the real part and b is the imaginary part.

(6 markah/marks)

- b) Bina satu kelas **Q4Main** untuk menguji aturcara. Aturcara akan meminta pengguna memasukkan bahagian nyata dan khayalan untuk dua nombor kompleks, membina dua objek **Complex** dan memaparkan nombor kompleks tersebut bersama hasil penambahan dan penolakan.

Create a **Q4Main** class to test the program. The program will ask the user to enter the real and imaginary parts of two complex numbers, create two **Complex** objects and display the complex numbers together with the addition and subtraction results.

Contoh output:

Sample output:

```

First complex number. Enter a number for the real part: 2<enter>
First complex number. Enter a number for the imaginary part: 6<enter>
Second complex number. Enter a number for the real part: 4<enter>
Second complex number. Enter a number for the imaginary part: 7<enter>

First complex number: (2 + 6i)
Second complex number: (4 + 7i)
Addition of the two complex numbers: (6 + 13i)
Subtraction of the two complex numbers: (-2 + -1i)

```

(Salin fail **Complex.java** dan **Q4Main.java** ke direktori akaun peperiksaan anda. Namakan semula sebagai **[matricNumberComplex.java; contoh: WIF160001Complex.java]** dan **[matricNumberQ4.java; contoh: WIF160001Q4.java]**)

(Copy the **Complex.java** and **Q4Main.java** files to your exam account directory. Rename as **[matricNumberComplex.java; example: WIF160001Complex.java]** and **matricNumberQ4.java; example: WIF160001Q4.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 esei. Jumlah markah esei 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 esei, membina satu objek **Essay** dan memaparkan maklumat mengenai esei 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: **WIF160001 GradedActivity.java**, **WIF160001Essay.java** dan **WIF160001Q5.java**]).

(Copy the **GradedActivity.java**, **Essay.java** and **Q5Main.java** files to your exam account directory. Rename as **[matricNumberComplex.java**, **matricNumberEssay.java**, **matricNumberQ5.java**; example: **WIF160001Complex.java**, **WIF160001Essay.java** and **WIF160001Q5.java**]).

(3 markah/marks)

TAMAT
END