CONFIDENTIAL

UNIVERSITI TEKNOLOGI MALAYSIA
FACULTY OF COMPUTING

TEST 1

SEMESTER II 2017/2018

SUBJECT CODE	**: SCSJ2154**
SUBJECT NAME	**: OBJECT ORIENTED PROGRAMMING**
YEAR/COURSE	**: 2 (SCSJ / SCSV / SCSB / SCSR)**
TIME	**: 8.00 pm – 10.00 pm (2 Hours)**
DATE	**: 28 March 2017 (Wednesday)**
VENUE	**: MPK1-10 (Block N28)**

INSTRUCTIONS TO THE STUDENTS:

- Read the problem and instructions carefully.
- References to any resources by any means except **OOP Lab Module** are strictly prohibited.
- You are given **TWO HOURS** to complete the test inclusive of the submission of your program.
- You must answer all the questions.
- You can download the java file for **Question 1** and input file for **Question 2** via UTM's e-learning system.
- Both of your programs must follow the input and output as shown in the examples.

SUBMISSION PROCEDURE:

- Only the source code (*i.e.* the file with the extension **.java**) is required for the submission.
- Submit the source code via the **UTM's e-learning system**.

This question booklet consists of 8 pages inclusive of the cover page.

QUESTION 1 – ERROR DEBUGGING **(40 Marks)**

You are given Program 1 (**`Test1.java`**) with syntax and/ or logical errors. The program consists of two classes: **`Test1`** and **`Subject`**. The program can be used to calculate Grade Point Average (GPA). The GPA is calculated by dividing the total amount of grade points earned by the total amount of credit hours attempted. Table 1 shows the grade point for each grade.

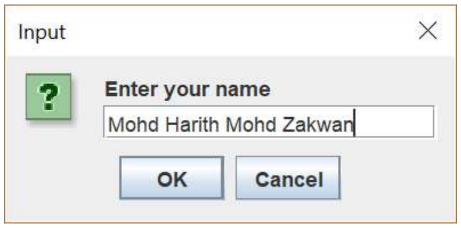
Table 1: Grade points

Grade	Grade Point
A	4.0
B	3.0
C	2.0
D	1.0
E	0.0

```java
//Program 1

import java.swingx.JOptionPane;
import java.util.Scanner;

class Subject
{
   private String code, name, grade;
   private int credit;
   int static totalCredit = 0;

   public Subject (String code, String name, int credit, String
                   grade)
   {
         this.code = code;
         this.name = name;
         this.credit = credit;
         this.grade = grade;
         totalCredit += credit;
   }

   public void getCode()
   {
         return code;
   }

   public void getName()
   {
         return name;
   }

   public void getGrade()
   {
         return grade;
   }

   public void getCredit()
   {
         return credit;
   }
}

public class Test1
{
   public double static getGradeValue (String grade)
   {
            if (grade == "A")
                  return 4.0;
            else
            if (grade == "B")
                  return 3.0;
            else
            if (grade == "C")
                  return 2.0;
            else
            if (grade == "D")
                  return 1.0;
            else
                  return 0.0;
   }

    public static void main(String args)
   {
         String studName, numSubj, codeSubj, nameSubj, grade,
          creditStr;
         int numSubject, credit;
         float totalValue = 0;

         Scanner inp = new Scanner();

         studName = JOptionPane.showInputDialog("Enter your name");
         numSubj = JOptionPane.showInputDialog("The number of
                  subject taken");
         JOptionPane.showMessageDialog(studName + " takes " +
          numSubj + " subject(s)", "Subject Info" +
          JOptionPane.INFORMATION_MESSAGE);


         numSubject = Integer.parseInteger(numSubj);
         Subject [] subj = new [numSubject] Subject;
         System.out.println("Please enter the data for your subject:
                           ");

         for (int i = 0; i < numSubject; i++)
         {
               System.out.println("\nSubject[" + (i + 1)+ "]");
               System.out.print("\tCode  : ");
               codeSubj = inp.nextLine();
               System.out.print("\tName  : ");
               nameSubj = inp.nextLine();
               System.out.print("\tCredit: ");
               creditStr = inp.nextLine();
               credit = Integer.parseInteger(creditStr);
               System.out.print("\tGrade : ");
               grade = inp.nextLine().toUppercase();
               subj[i] = Subject(codeSubj, nameSubj, credit, grade);
         }

         System.out.println("\n\n\nRESULT FOR SEM 2, 2017/2018");
         System.out.println("\nNAME: " + studName.toUppercase());
         for (int i = 0; i < numSubject; i++)
         {
               System.out.print("\n%-5d%-12s%-35s%-5s%.2f", i + 1,
                subj[i].getCode(), subj[i].getName(),
                subj[i].getGrade(), getGradeValue(getGrade()));
               totalValue += getGradeValue(subj[i].getGrade()) *
                subj[i].getCredit();
         }
         System.out.println("\n\nTOTAL CREDITS = " + totalCredit);
         System.out.print( "YOUR GPA      = %.2f\n\n", totalValue/
          totalCredit);
     }
 }
```

Debug the errors, then compile and run the program. The program should produce the following output:

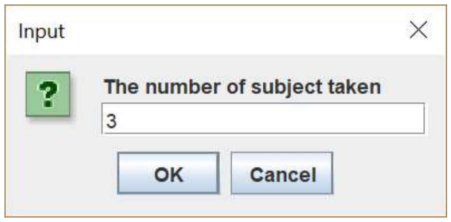
a) Input dialog box as shown in Figure 1. Enter your name in the textbox. Click on the OK button.



Enter your name
Mohd Harith Mohd Zakwan

Figure 1

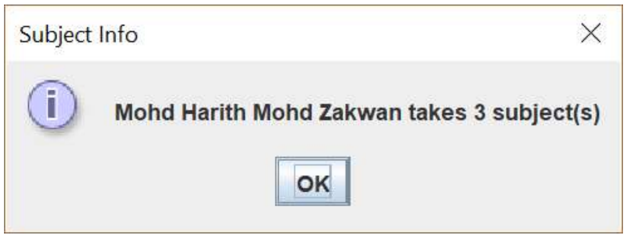
b) Then the input dialog box as shown in Figure 2 will be displayed. Enter the number of subject you have taken in the textbox. Click on the OK button.



The number of subject taken
3

Figure 2

c) Next, the message dialog box as shown in Figure 3 will be displayed. Click on the OK button.



Subject Info

Mohd Harith Mohd Zakwan takes 3 subject(s)

Figure 3

d) Then you need to enter the data for your subject (input data from keyboard). Figure 4 shows the example of input and output generated from this program. Note that the text in **bold** indicates input entered by the user.

```
Please enter the data for your subject:

Subject[1]
        Code  : SCSJ 2154
        Name  : Object Oriented Programming
        Credit: 4
        Grade : A

Subject[2]
        Code  : SCSJ 2203
        Name  : Software Engineering
        Credit: 3
        Grade : B

Subject[3]
        Code  : SCSV 1223
        Name  : Web Programming
        Credit: 3
        Grade : A


RESULT FOR SEM 2, 2017/2018

NAME: MOHD HARITH MOHD ZAKWAN

1    SCSJ 2154   Object Oriented Programming        A    4.00
2    SCSJ 2203   Software Engineering               B    3.00
3    SCSV 1223   Web Programming                    A    4.00

TOTAL CREDITS = 10
YOUR GPA      = 3.70

Press any key to continue . . .
```

Figure 4

After you get all the outputs as stated above, please submit your **successful** program named **`Test1.java`** via the UTM's e-learning system.

QUESTION 2 – PROBLEM SOLVING **(60 Marks)**

Given the following UML class diagram in Figure 5, write three complete Java programs, **`DailyReport.java`**, **`Burger.java`** and **`MenuList.java`** based on the instruction given in (a), (b) and (c).

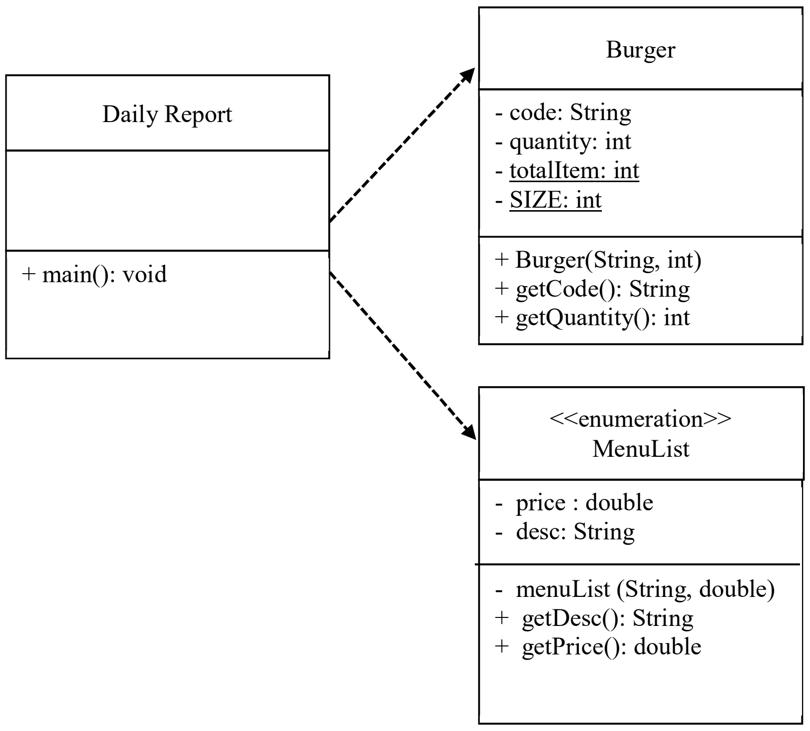


Figure 5: UML class diagram

a) Write a class **`Burger`** with the following information: (11 Marks)

(i) Define a constant named SIZE with the value 15.

(ii) Write a constructor for class **`Burger`** that initializes **`code`** and **`quantity`** instance variables through parameter passing. The constructor must be able to calculate the total number of items sold using **`static totalItem`** variable.

(iii) Write suitable code for the getter (accessor) methods.

b) Write a class **`MenuList`** with the following information: (12 Marks)

(i) The class uses **`enum`** data type.

(ii) The class has a fixed set of constants as listed in Table 1.

(iii) Write a constructor for class **`MenuList`** that initializes **`price`** and **`desc`** instance variables through parameter passing.

(iv) Write suitable code for the getter (accessor) methods.

Table 1: Set of constant for **`MenuList`** class

Code	Description	Price
B101	McChicken	8.90
B102	Fillet-O-Fish	8.90
B103	Cheeseburger	5.50
B201	Chicken McNuggets	11.50
B202	GCB	12.50
B203	Spicy Chicken McDeluxe	11.50
B204	Big Mac	10.90
B301	Double GCB	18.20
B302	Double Fillet-O-Fish	11.95
B303	Double Cheeseburger	9.50
B304	Double Spicy Chicken McDeluxe	17.25

c) Write a class **`DailyReport`** that only has **`main()`** method with the following codes: (37 Marks)

(i) Read an input file named **`Input.txt`** with a list of code and quantity of burger sold.

```
B101 45
B102 13
B103 25
B201 30
B202 9
B203 8
B204 13
B301 32
B302 28
B303 17
B304 39
```

(ii) Create an array of objects from class **`Burger`** to store the value that read in c(i).

(iii) Create an object from class **`MenuList`** to retrieve a description and price for burger based on burger's code.

(iv) Calculate the total price for each and whole burger sold based on quantity that you read in c(i) and price that you retrieve in c(iii).

(v) Display the total items (burgers) sold and the total daily income (based on the total price for whole burger sold that calculated in c(iv)).

The program should produce the output as shown in Figure 6.

```
                            ABC BURGER DAILY SALES REPORT

NUM  CODE   DESCRIPTION                      PRICE(RM)     QUANTITY     TOTAL PRICE(RM)
1    B101   McChicken                           8.90          45            400.50
2    B102   Filet-O-Fish                        8.90          13            115.70
3    B103   Cheeseburger                        5.50          25            137.50
4    B201   Chicken McNuggets                  11.50          30            345.00
5    B202   GCB                                12.50           9            112.50
6    B203   Spicy Chicken McDeluxe             11.50           8             92.00
7    B204   Big Mac                            10.90          13            141.70
8    B301   Double GCB                         18.20          32            582.40
9    B302   Double Filet-O-Fish                11.95          28            334.60
10   B303   Double Cheeseburger                 9.50          17            161.50
11   B304   Double Spicy Chicken McDeluxe      17.25          39            672.75

                        TOTAL ITEMS SOLD = 259 burger(s)
                        TOTAL INCOME     = RM 3096.15



Press any key to continue . . .
```

Figure 6: Output