CONFIDENTIAL

SCHOOL OF COMPUTING
Faculty of Engineering

UNIVERSITI TEKNOLOGI MALAYSIA

TEST 1

SEMESTER II 2018/2019

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	**: 19 March 2019 (Tuesday)**
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 files 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 7 pages inclusive of the cover page.

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

You are given Program 1 (**BankAccount.java**) with syntax and/ or logical errors. The program consists of two classes: **Account** and **BankAccount**. The program can be used to show bank account information where it can calculate Annual Interest Rate for monthly rate. The method ***getMonthlyInterest()*** is to return monthly interest, not the interest rate. Monthly interest is ***balance * monthlyInterestRate***. ***monthlyInterestRate*** is ***annualInterestRate / 12***. ***Note: annualInterestRate*** is a percentage, for example 4.5%. You need to divide it by 100.

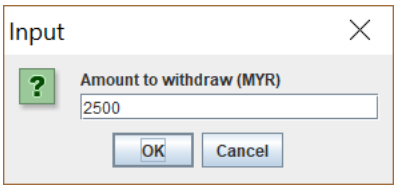
```
1   import javax.swing.JOptionPane;
2   import java.util.Date;
3
4   class Account {
5     private int id;
6     private double balance;
7     private static double annualInterestRate;
8     private int dateCreated;
9
10    Account() {
11      id = 0;
12      balance = 0;
13      annualInterestRate = 0;
14      dateCreated = new Date();
15    }
16
17    Account(int newId, double newBalance) {
18      id = newId;
19      balance = newBalance;
20      dateCreated = new Date();
21    }
22
23    public void setId(int newId) {
24      id = newId;
25    }
26
27    public void setBalance(char newBalance) {
28      balance = newBalance;
29    }
30
31    public void setAnnualInterestRate(char newAnnualInterestRate) {
32      annualInterestRate = newAnnualInterestRate;
33    }
34
35    public int getId() {
36      return id;
37    }
38
39    public double getBalance() {
40      return balance;
41    }
42
43    public double getAnnualInterestRate() {
44      return annualInterestRate;
45    }
46
47    public String getDateCreated() {
48      return dateCreated.toString();
49    }
50
51    public double getMonthlyInterestRate() {
52      return;
```

```
53    }
54
55    public double getMonthlyInterest() {
56      return balance * (getMonthlyInterestRate() / 100);
57    }
58
59    public void withdraw(double amount) {
60      balance -= amount;
61    }
62
63    public void deposit(double amount) {
64      balance += amount;
65    }
66  }
67
68  public class BankAccount {
69    public static void main(String [] args)
70    {
71      String wDraw;
72      double w;
73
74      Account account1 = new Account();
75      account.setAnnualInterestRate();
76      wDraw = JOptionPane.showMessageDialog("Amount to withdraw (MYR)");
77      w = Double.nextDouble();
78      account.withdraw();
79      account.deposit(3000);
80
81      System.out.println("----------------------------------------");
82      System.out.println("\n          Account Statement");
83      System.out.println("----------------------------------------");
84      System.out.println("Account ID: " + account.getId());
85      System.out.println("Date created: " + account.get());
86      System.out.printf("Balance: MYR%.2f\n" + account.get());
87      System.out.printf("Monthly interest: MYR%.2f\n" + account.get());
88      System.out.println("----------------------------------------");
89
90      Account account2 = new Account();
91      account.setAnnualInterestRate();
92      account.withdraw();
93      account.deposit(300);
94
95      System.out.println("----------------------------------------");
96      System.out.println("\n          Account Statement");
97      System.out.println("----------------------------------------");
98      System.out.println("Account ID: " + account.getId());
99      System.out.println("Date created: " + account.get());
100     System.out.printf("Balance: MYR%.2f\n" + account.get());
101     System.out.printf("Monthly interest: MYR%.2f\n" + account.get());
102     System.out.println("----------------------------------------");
103   }
104 }
```

Debug the errors, then compile and run the program by using the following information:

Account 1

a) Creates Account 1 with an account ID of 1122.
b) Balance of MYR20,000, and an annual interest rate of 4.5%.
c) Enter the amount to withdraw in the input dialog box.



d) Use the withdraw method to withdraw the amount that you entered in (c).
e) Use the deposit method to deposit MYR 3,000.

Account 2
a) Creates Account 2 with an account ID of 1144.
b) Balance of MYR5,000, and an annual interest rate of 4.5%.
c) Use the withdraw method to withdraw MYR250.
d) Use the deposit method to deposit MYR300.

Print the balance of Account 1 and Account 2, the monthly interest, and the date when this account was created. Figure below shows the example of input and output generated from this program.

```
----------------------------------------

          Account Statement
----------------------------------------
Account ID: 1122
Date created: Tue Mar 19 08:35:27 SGT 2019
Balance: MYR20500.00
Monthly interest: MYR76.88
----------------------------------------
----------------------------------------

          Account Statement
----------------------------------------
Account ID: 1144
Date created: Tue Mar 19 08:37:46 SGT 2019
Balance: MYR5050.00
Monthly interest: MYR18.94
----------------------------------------
Press any key to continue . . .
```

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

Write three complete Java programs, **`StudentResults.java`**, **`Course.java`** and **`Student.java`** to generate Examination Result Slip based on the instruction given in (a), (b) and (c).

a) Write a class **`Student`** with the following information: (8.5 Marks)

(i) Private member attributes: **name**, **matricsNo**, and **GPA** (Grade Point Average).

(ii) Static instance variable: **totalCredit** (total credit earned).

(iii) A default constructor that initializes static **totalCredit** variable to **0** (zero).

(iv) An appropriate getter (accessor)/ setter (mutator) methods.

b) Write a class **`Course`** with the following information: (15.5 Marks)

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

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

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

Code	Description	Credit
SCSJ2154	Object Oriented Programming	4
SCSD2623	Database Programming	3
SCSV1223	Web Programming	3
SCSJ2203	Software Engineering	3
SCSR2043	Operating Systems	3
ULAB2122	Advanced Academic English Skills	2
UCSD2762	Fundamentals of Technopreneurship	2
SCSD3761	Technopreneurship Seminar	1

(iii) Private member attributes: **credit**, **description**, and **grade**.

(iv) A constructor that initializes **credit** and **description** instance variables through parameter passing.

Table 2: Grade points

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

(v) An appropriate getter (accessor)/ setter (mutator) methods.

(vi) A method named **getGradeValue** that return a value (grade point) of grade. Table 2 shows the grade point for each grade.

c) Write a class **`StudentResults`** that only has **`main()`** method with the following codes: (36 Marks)

(i) Read a matrics number. The input entered by the user (input data from keyboard), i.e. ***A17CS0010***.

(ii) By using matrics number in c(i), read an input file named ***`matricsNumber.txt`*** (i.e. ***A17CS0010.txt***) that consist of student's name and a list of grade and course code. Three input files are provided named **`A17CS0001.txt`**, **`A17CS0010.txt`**, and **`A17CS0100.txt`** to help you to test/ execute your program. Figure 1 shows the example of two input files (for two different matrics numbers).

```
Amir Hafsyam bin Abdullah
A SCSJ2203
B SCSJ2154
A ULAB2122
B SCSR2043
A SCSV1223
```

(a) A17CS0010.txt

```
Abdullah Abu bin Hanifah
A SCSJ2154
B SCSJ2203
A ULAB2122
C SCSR2043
A SCSV1223
B UCSD2762
```

(b) A17CS0100.txt

Figure 1: Example of input files

(iii) Create an object from class **`Student`** to store the value that you read in c(i) and c(ii).

(iv) Create an object from class **`Course`** to retrieve a description and credit for course based on course's code.

(v) Calculate the total credit earned based on credit that you retrieve in c(iv).

(vi) Calculate the Grade Point Average (GPA). The GPA is calculated by dividing the total amount of grade points earned by the total amount of credit earned.

(vii) Display the Examination Result Slip for the student in the output file named **`Slip`*`matricsNumber`*`.txt`** (i.e. ***SlipA17CS0010.txt***). The program should produce the output as shown in Figure 2.

(viii) Figure 3 shows an example run of program. Note that, **bold** texts indicate keyboard input entered by the user.

```
EXAMINATION RESULTS FOR SEM 2, 2018/2019

NAME       : AMIR HAFSYAM BIN ABDULLAH
MATRICS NO : A17CS0010

CODE        COURSE                                  CREDIT    GRADE
============================================================
SCSJ2203    Software Engineering                      3         A
SCSJ2154    Object Oriented Programming               4         B
ULAB2122    Advanced Academic English Skills          2         A
SCSR2043    Operating Systems                         3         B
SCSV1223    Web Programming                           3         A
============================================================

CREDIT EARNED : 15
GPA           : 3.53
```

(a) SlipA17CS0010.txt

```
EXAMINATION RESULTS FOR SEM 2, 2018/2019

NAME       : ABDULLAH ABU BIN HANIFAH
MATRICS NO : A17CS0100

CODE        COURSE                                  CREDIT    GRADE
============================================================
SCSJ2154    Object Oriented Programming               4         A
SCSJ2203    Software Engineering                      3         B
ULAB2122    Advanced Academic English Skills          2         A
SCSR2043    Operating Systems                         3         C
SCSV1223    Web Programming                           3         A
UCSD2762    Fundamentals of Technopreneurship         2         B
============================================================

CREDIT EARNED : 17
GPA           : 3.35
```

(b) SlipA17CS0100.txt

Figure 2: Example of output files

```
EXAMINATION RESULT SLIP SYSTEM

Enter student's matrics number: A17CS0010
Examination Slip is generated...

Press any key to continue . . .
```

Figure 3: Example run of program