SCHOOL OF COMPUTING
Faculty of Engineering

SESSION 2020/2021 SEMESTER 2

SECJ2154-11 OBJECT ORIENTED PROGRAMMING

Dr. NORSHAM BINTI IDRIS

ASSIGNMENT 2 (INDIVIDUAL)

SUBJECT CODE: SECJ2154

YEAR/COURSE: 2SECV

Simon Chong Kai Yuen
A19EC3028
simonnchong@gmail.com

class Person

```
//Simon

// a)i)  4 attributes: gender, age, background and status
public class Person {
	private String gender;
	private int age;
	private String background;
	private String status;
	static int counter;

	// a)ii) constructor for class Person
	Person(String g, int a, String b, String s){
		gender = g;
		age = a;
		background = b;
		status = s;
	}

	// a)iii) getter (accessor) methods.
	public String getGender() {
		return gender;
	}

	public int getAge() {
		return age;
	}

	public String getBackground() {
		return background;
	}

	public String getStatus() {
		return status;
	}

	public String toString() {
	    return String.format("%-10s %-8s %-17s %-1s", gender, age,
background, status);
	}
}
```

class CaseList

```
//Simon

// b)i) class uses enum data type
public enum CaseList {
	CASE1("TRAVEL ABROAD", "QUARANTINE"),
	CASE2("CLOSE CONTACT", "QUARANTINE"),
	CASE3("COVID SYMPTOM", "WARDED"),
	CASE4("CRITICAL", "ICU"),
	CASE5("DISCHARGED", "RECOVERED");

	// b)ii) has 2 attributes: background and status
	private String background;
	private String status;

	// b)iii) constructor for class CaseList
	CaseList(String b, String s){
		background = b;
		status = s;
	}

	// b)iv) getter (accessor)methods
	public String getBackground() {
		return background;
	}

	public String getStatus(){
		return status;
	}
}}
```

class DailyReport (main)

```
//Simon

import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Scanner;

public class DailyReport {

     public static void main(String[] args) throws
FileNotFoundException{

           // c)ii)  arraylist of objects from class Person
           ArrayList<Person> arrList = new ArrayList<Person>();

           File file = new File("InputSB.txt");
           Scanner inputFile= new Scanner(file);

           int q=0, w=0, r=0, i=0; // q=quarantine, w=warded,
r=recovered, i=ICU, theses variables are used to count the cases
           String b = null, s = null; // b=background, s=status,
these variables are use to check the background and status from case
           String g = "";

           // c)i) Read an input file named InputSB.txt with a list
of gender, age and Person’s Case
           while (inputFile.hasNext()) {

                 //this will read the string from the .txt file and
assign it to a variable, reading will be stop by a space
                String gender = inputFile.next();
                int age = inputFile.nextInt();
                String _case = inputFile.next();

                //use to display the short-form of gender to full
word
                switch(gender) {
                case "F":
                  g = "FEMALE";
                  break;
                case "M":
                  g = "MALE";
                  break;
                }
```

```
                // c)iv) Count how many cases for Quarantine, Warded,
Recover, in ICU
                // also checking what is the CASE from the text and
read the message from enum class
                if(_case.equals(String.valueOf(CaseList.CASE1))) {
                  b = CaseList.CASE1.getBackground();   //set the
CASE1 background message from enum class
                  s = CaseList.CASE1.getStatus(); //set the CASE1
status message from enum class
                  q++; //to count the number of CASE1, incremental by
1 for every reading from .txt file
                  }
                else if(_case.equals(String.valueOf(CaseList.CASE2)))
{
                  b = CaseList.CASE2.getBackground(); //set the CASE2
background message from enum class
                  s = CaseList.CASE2.getStatus(); //set the CASE2
status message from enum class
                  q++; //to count the number of CASE2, incremental by
1 for every reading from .txt file
                }
                else if(_case.equals(String.valueOf(CaseList.CASE3)))
{
                  b = CaseList.CASE3.getBackground(); //set the CASE3
background message from enum class
                  s = CaseList.CASE3.getStatus(); //set the CASE3
status message from enum class
                  w++; //to count the number of CASE3, incremental by
1 for every reading from .txt file
                }
                else if(_case.equals(String.valueOf(CaseList.CASE4)))
{
                  b = CaseList.CASE4.getBackground(); //set the CASE4
background message from enum class
                  s = CaseList.CASE4.getStatus(); //set the CASE4
status message from enum class
                  i++; //to count the number of CASE4, incremental by
1 for every reading from .txt file
                }
                else if(_case.equals(String.valueOf(CaseList.CASE5)))
{
                  b = CaseList.CASE5.getBackground(); //set the CASE5
background message from enum class
                  s = CaseList.CASE5.getStatus(); //set the CASE5
status message from enum class
                  r++; //to count the number of CASE5, incremental by
1 for every reading from .txt file
                }
```

```
                Person person = new Person(g, age, b, s); //create
object from class Person and use its constructor to pass the text
read from .txt
                arrList.add(person); //save object person into
arrList which is an object from ArrayList

                // c)v) Count the number of all cases by using static
variable in class Person
                Person.counter++;
            }

            inputFile.close();

            // c)iii) Print all cases in the CaseList enum class
            System.out.println("\nCovid-19 Cases and Action
Required");
            for(CaseList case_list : CaseList.values()) {
                 System.out.printf("%-4s%-1s %-17s %-15s \n",
case_list, ":", case_list.getBackground(), case_list.getStatus());
                 }

            //print the header of the report
            System.out.println("\n\t\tCOVID-19 DAILY REPORT");
            System.out.printf("%-6s %-10s %-8s %-18s%-1s","Case",
"Gender", "Age", "Background", "Status\n");

            int x = 0; //use to count the case numbering
            for(Person personList : arrList) {
                 System.out.printf("%-7s", ++x);  //use to print the
case numbering
                 System.out.println(personList.toString()); //print
all element from the stored ArrayList
            }

            //calculate the number of corresponding case and the
total number of cases.
            System.out.println("\nTotal In Quarantine = " + q);
            System.out.println("Total In ICU = " + i);
            System.out.println("Total Warded = " + w);
            System.out.println("Total Recovered = " + r);
            System.out.println("Total Cases = " + Person.counter);
      }
}
```

Output

```
Covid-19 Cases and Action Required
CASE1: TRAVEL ABROAD     QUARANTINE
CASE2: CLOSE CONTACT     QUARANTINE
CASE3: COVID SYMPTOM     WARDED
CASE4: CRITICAL          ICU
CASE5: DISCHARGED        RECOVERED

                COVID-19 DAILY REPORT
Case   Gender     Age      Background        Status
1      MALE       67       TRAVEL ABROAD     QUARANTINE
2      MALE       76       CLOSE CONTACT     QUARANTINE
3      MALE       50       COVID SYMPTOM     WARDED
4      FEMALE     55       COVID SYMPTOM     WARDED
5      MALE       60       CRITICAL          ICU
6      MALE       32       CLOSE CONTACT     QUARANTINE
7      FEMALE     45       TRAVEL ABROAD     QUARANTINE
8      FEMALE     58       COVID SYMPTOM     WARDED
9      MALE       24       TRAVEL ABROAD     QUARANTINE
10     FEMALE     59       CLOSE CONTACT     QUARANTINE
11     MALE       78       CRITICAL          ICU
12     MALE       48       DISCHARGED        RECOVERED
13     FEMALE     62       DISCHARGED        RECOVERED
14     MALE       65       DISCHARGED        RECOVERED

Total In Quarantine = 6
Total In ICU = 2
Total Warded = 3
Total Recovered = 3
Total Cases = 14
```