CONFIDENTIAL

School of Computing

UNIVERSITI PENYELIDIKAN

UNIVERSITI TEKNOLOGI MALAYSIA

TEST 2

SEMESTER 11 2018/2019

SUBJECT CODE	**:**	**SCSJ2154**
SUBJECT NAME	**:**	**OBJECT ORIENTED PROGRAMMING**
YEAR/PROGRAM	**:**	**2 (SCSJ / SCSV / SCSB / SCSR)**
TIME	**:**	**8:00 PM – 10:00 PM (2 HOURS)**
DATE	**:**	**16th April 2019**
VENUE	**:**	**MPK 1 – 10 (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 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.
- Archive the source code as a single .zip file if there are more than single .java files involved.
- Submit the source code / archived file via the UTM's e-learning system.

QUESTION 1 – ERROR DEBUGGING (40 Marks)

You are given Program 1 (`Medic.java`) with syntax errors. The program consists of three classes: `Hospital`, `Doctors` and `Birthday`. The program can be used to show doctors information such as doctor names, hospital name and doctor's birthday.

```
import java.util.Array_List;

public class Medic {

 public static void main(String[] args) {
  Hospital hosp = new Hospital();
  hosp.setHospitalName("HSA");

  Doctors doc1 = new Doctors(1980,12,1);
  doc1.setDoctorName("Dr.Abu Ali");
  Doctors doc2 = new Doctors(1981,1,1);
  doc2.setDoctorName("Dr.Kim Lee");
  Doctors doc3 = new Doctors(1985,12,12);
  doc3.setDoctorName("Dr.Nadela A/P Ram");

  Array_List<Doctors> doctorList = new Array_List;
  doctorList.Add(doc1);
  doctorList.Add(doc2);
  doctorList.Add(doc3);


  hosp.setDoctor();

  System.out.println("Size of list: " + Doctors.size);
  System.out.println(getDoctors +"are Doctors of"+  getHospName);


  doctorList.removes(1);
  System.out.println("Display list:" + Doctors );

  Doctors newDoc1= new Doctors(1984,5,8);
  newDoc1.setDoctorName("Dr.Badariah");
  doctorList.Add(0,newDoc1);
  Doctors newDoc2= new Doctors(1982,22,5);
  newDoc2.setDoctorName("Dr.Zang Yu");
  doctorList.Add(3,newDoc2);

 System.out.println("Display list: "+ Doctors);

 System.out.println("Doctor list and their birthday: ");
 System.out.println("First doctor: "+get()+ getBday());
 System.out.println("Second doctor: "+get()+getBday());
 System.out.println("Third doctor : "+get()+getBday());
 System.out.println("Forth doctor "+get()+getBday());

 }

} //Medic
```

```
class Hospital {
     private String hospitalName;
        ArrayList<> doctors;

    public String getHospName() {
        return hospitalName;
    }
     public void setHospitalName(String hospName) {
        hospitalName = hospName;
    }
    public ArrayList<Doctor> getDoctors() {
        return doctors;
    }
    public void setDoctor(ArrayList<Doctor> doc) {
        doctors = doc;
    }

} //Hospital

class Doctors {
    private String doctorName;
    private bday;

    Doctors(int y,int m,int d){
        bday=new Birthday(y, m, d);
    }

    public String getDoctorName() {
        return doctorName;
    }

    public void setDoctorName(String docName) {
        doctorName = docName;
    }

    public String toString() {
        return doctorName;
    }

   public Birthday getBday() {
    return ;}
} //Doctors

class Birthday{
        int year,month,day;

        Birthday(int y,int m,int d){
            year=y;
            month=m;
            day=d;
        }

        public String toString(){
           return String.format("%s-%s-%s", year,month,day);
        }
    }//Birthday
```

Program 1: Hospital, Doctors and Birthday class relationship

Debug the errors, then compile and run the program by using the given information in class Medic.

Name	Birthday
Dr.Abu Ali	1980,12,1
Dr.Kim Lee	1981,1,1
Dr.Nadela A/P Ram	1985,12,12
Dr.Badariah	1984,5,8
Dr.Zang Yu	1982,22,5

Figure below shows the output generated from this program:

```
Size of list:3
[Dr.Abu Ali, Dr.Kim Lee, Dr.Nadela A/P Ram]are Doctors of HSA
Display list: [Dr.Abu Ali, Dr.Nadela A/P Ram]
Display list: [Dr.Badariah,Dr.Abu Ali, Dr.Nadela A/P Ram,Dr.Zang Yu]
Doctor list and their birthday:
First doctor: Dr.Abu Ali 1980,12,1
Second doctor: Dr.Nadela A/P Ram 1985,12,12
Third doctor : Dr.Badariah 1984,5,8
Forth doctor: Dr.Zang Yu 1982,22,5
```

QUESTION 2 – PROBLEM SOLVING (60 Marks)

Figure 1 is a class diagram that represents a complete definition and relationship of `Person`, `Customer`, `Item`, `Material`, `Service`, `TestItem` and `TestCustomer` classes.

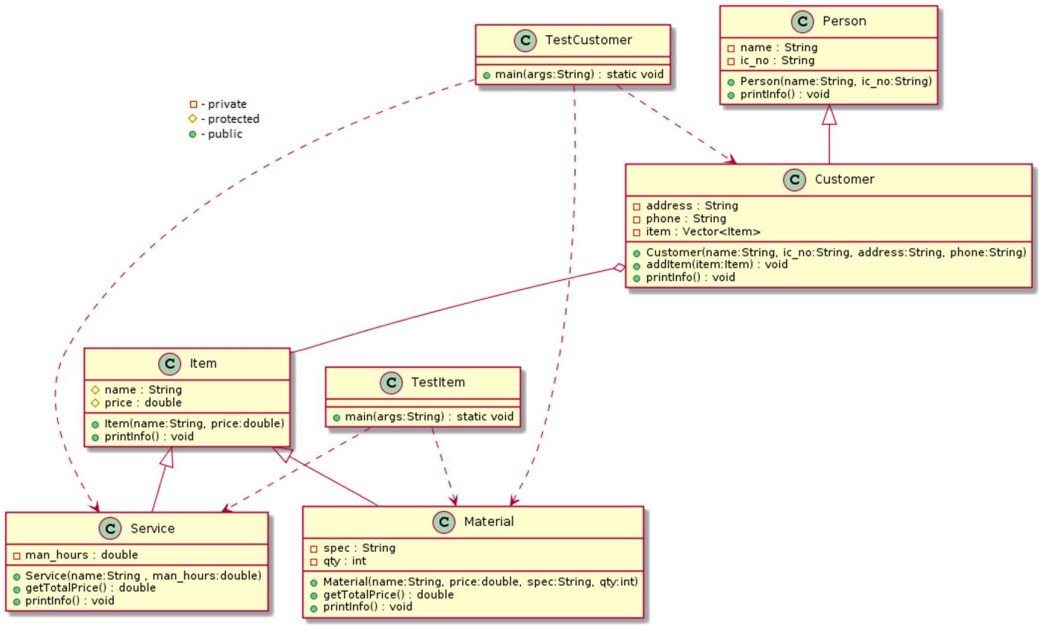

Figure 1: Class diagram of `Person`, `Customer`, `Item`, `Material`, `Service`, `TestItem` and `TestCustomer` classes.

Two java programs (`TestItem.java` and `TestCustomer.java`) have been prepared to test the `Item` and `Customer` classes. Compile and run these two test programs before you proceed with the main tasks given in this question. Below is the default output of these test programs:

Default output of `TestItem.java`:

```
Test 'Material' class:
----------------------
Item Name: Engine Oil
    Price: 130.8

Test 'Service' class:
---------------------
Item Name: Engine Oil Service
    Price: 15.0

---------------------
Item Name: Major Service
    Price: 15.0
```

Default output of `TestCustomer.java`:

```
Test 'Customer' class:
----------------------
   Name: Alice
 IC No.: 456
```

Based on the class diagram given in Figure 1, do the following two tasks:

a) Complete the implementation of `Material.java` and `Service.java` source files so the `TestItem.java` program can produce the output as shown below:

```
Test 'Material' class:
----------------------
Item Name: Engine Oil
    Price: 130.8
       Type: MATERIAL
      Spec.: Semi Synthetic 15W/40
   Quantity: 2
Total Price: 261.6

Test 'Service' class:
---------------------
Item Name: Engine Oil Service
    Price: 15.0
       Type: SERVICE
  Man Hours: 0.5
Total Price: 15.0

---------------------
Item Name: Major Service
    Price: 15.0
       Type: SERVICE
  Man Hours: 2.8
Total Price: 49.5
```

As a guide, below is the list of implementations need to be done inside the `Material.java` and `Service.java` files:

- Define class/instance variable
- Complete the class's constructor implementation
- Implement `getTotalPrice` method
- Implement `printInfo` method

Inside the `Material` and `Service` classes, the `getTotalPrice()` method should behaves differently as follows:

- `Material` class - Total price is simply `qty * price`
- `Service` class - The rate of total price is fixed at RM 15 if `man_hours` is not more than 0.5 hours, otherwise it is rated by using the formula: `15 + 15 * (man_hours - 0.5)`

(30 Marks)

b) Complete the implementation of `Customer.java` source file so the `TestCustomer.java` test program can produce the output as shown below:

```
Test 'Customer' class:
----------------------
   Name: Alice
 IC No.: 456
Address: No. 6, KTC UTM, Johor
  Phone: 011-7123987

Ordered Item(s):
................

Item Name: Engine Oil
    Price: 130.8
       Type: MATERIAL
      Spec.: Semi Synthetic 15W/40
   Quantity: 2
Total Price: 261.6

Item Name: Oil Filter
    Price: 18.3
       Type: MATERIAL
      Spec.: Saga OEM
   Quantity: 2
Total Price: 36.6

Item Name: Engine Oil Service
    Price: 15.0
       Type: SERVICE
  Man Hours: 0.8
Total Price: 19.5

GRAND TOTAL PRICE: 317.70
```

Below is the list of implementations need to be done inside the `Customer.java` file:

- Define class/instance variable
- Complete the class's constructor implementation
- Complete the implementation of `addItem` method
- Add method to print customer info and items ordered

(30 Marks)