St Edward's College Malta
Mid-Year Examinations   February 2020

Year 13   Computing HL   Time: 1 hour 30 mins
Paper 2

Name and Surname

Instructions to Students:

1. Do not open this examination paper until instructed to do so.
2. Write your name and surname on this page.
3. Read all instructions and questions carefully.
4. Answer ALL questions.
5. Diagrams must be drawn in pencil.
6. Leave the last 10 minutes for revision of paper.

For teacher's use only

Mr E. Attard Cassar

Question	1a	1b	1c	1d	1e	2a	2b	2c	3a	3b	4a	4b	4c	4d	Total
Obtained															
Allotted	1	4	2	6	8	7	6	5	2	4	6	6	4	4	65

(Option D — Object-oriented programming)

A hotel has 100 rooms and uses an OOP program to deal with allocating the rooms to clients when they arrive at the hotel. The program contains a **Client** class and a **Room** class. A user-defined **Dates** class is also used. Part of the **Client** class and **Dates** class are shown below.

```
public class Client
{
    private int customerID;
    private String name;
    private Dates arrive;
    private Dates leave;
    private Room bedroom;
    public Client(int id, String c, Dates dateIn, Dates dateOut, Room r)
    {
        setCustomerID(id);
        setName(c);
        setArrive(dateIn);
        setLeave(dateOut);
        setBedroom(r);
    }
    public void setCustomerID(int id) {customerID = id;}
    public void setName(String c) {name = c;}
    public void setArrive(Dates dateIn) {arrive = dateIn;}
    public void setLeave(Dates dateOut) {leave = dateOut;}
    public void setBedroom(Room r) {bedroom = r;}
    public int getCustomerID() {return customerID;}
    public String getName() {return name;}
    public Dates getArrive() {return arrive;}
    public Dates getLeave() {return leave;}
    public Room getBedroom() {return bedroom;}
    public void bill()
    {
```

```java
        ... // method that calculates the bill for this client
    }
}


public class Dates
{
    private int day;
    private int month;
    private int year;
    public Dates(int day, int month, int year)
    {
        this.day = day;
        this.month = month;
        this.year = year;
    }
    public int getDay() {return day;}
    public int getMonth() {return month;}
    public int getYear() {return year;}
    public static int StayDays(Dates x, Dates y)
    {
        ... // method that calculates the number of nights between x and y
    }
}
```

The Room class contains the following variables.

- **roomNumber** a value that identifies the room

- **beds** the number of beds in the room

- **price** the price of the room, per night

- **empty** indicating whether or not the room is occupied.

1.

    a. State the relationship between **Client** and **Room**.   [1]

b. Construct a UML diagram for the **Room** class. [4]

c. Outline one advantage of using a class to represent the dates. [2]

The **Room** objects are held in the array **allRooms**[] in ascending order of **roomNumber**. For example, the object for room 5 is held in **allRooms[4].**

d. Construct a method, **findRooms()**, that searches **allRooms[]** and returns the **roomNumber** of all empty rooms that have two beds. [6]

e. Construct the method, **bill()**, in the **Client** class to calculate and output a bill for a client based on the price of the room per night and the number of nights spent in the hotel.

The bill should include:

- the client's name
- the room number
- the date that they arrived
- the date that they are leaving
- the total number of nights they stayed
- the total cost. [8]

2. The hotel accepts group bookings where many clients, such as tour groups, can be allocated rooms on arrival. The **Group** class contains the name of the group, the number of rooms used by the group and a method **bill()** that calculates the total bill for the group.

Part of the **Group** class is shown below.

```
public class Group
{
    private String name; // name of group
    private int number; // number of rooms allocated to the group
    public Group(String name, int number)
    {
        this.name = name;
        this.number = number;
    }
    public String getName() {return name;}
    public int getNumber() {return number;}
    int[] gRooms = new int[number];
```

```
            // array to hold room numbers allocated to the group
            public double bill(int[] gRooms)
            {
                  ... // method that calculates the bill for the group
            }
      }
```

The **GClient** class represents a client who is part of a group. It inherits all the class members of the Client class and also stores the name of the group.

    a. Construct the **GClient** class. [7]

    b. With reference to the objects, methods and parameters, describe the process of allocating rooms when a tour group called "Happy Travellers" arrives at the hotel and is allocated 15 rooms, each of which has two beds. [6]

    c. Construct the method, **bill(int[] gRooms)**, to calculate and output the total cost of the group's rooms for one day. [5]

3. The company wishes to update this OOP program and sell it to other hotels. They hire a programmer to make appropriate changes to the program.

    a. Outline the responsibilities that the programmer has when updating the program. [2]

    b. Discuss the features of modern programming languages that enable the program to be sold in other countries. [4]

4. Clients can book over the internet and are allocated a room on arrival. When a booking is made, a **Client** object is added to a linked list, **Bookings**, which is linked in order of the client's arrival time, stored in **arrive**.

If any **Client** objects have the same arrival date they are ordered so that the most recently added object will appear first in the linked list.

The **Dates** class contains two methods to compare dates:

- public static Dates compareDate(Dates x, Dates y) which returns the earlier of the two dates

- public static Boolean equalDate(Dates x, Dates y) which returns true if the dates are the same.

In answering parts (a) and (b), you should use the methods of the **LinkedList** class.

a. Construct a method, **newClient()**, to insert a Client object into the correct position in **Bookings**. [6]

On the morning of each day, the hotel staff requires a list of clients who are expected to arrive that day.

b. Construct a method **todayClients(Dates today)** which, for all clients arriving today:

- removes their **Client** object from **Bookings**
- places their **Client** object in an array **arrivalsToday[]**
- sorts the array in order of the date that they will leave the hotel.

Note: You can assume that bookings for days previous to today, have already been removed from **Bookings** [6]

c. Discuss how the use of library classes simplifies the construction of the methods in parts (a) and (b). [4]

Alternatively, the **Client** objects of clients arriving today (05/05/2016) could be held in a binary tree in order of **dateOut**

d. By drawing such a binary tree for the clients in the following table, who arrived today, explain the recursive way in which the tree can be traversed using an inorder traversal. [4]

customerID	dateOut
121	08/05/2016
154	14/05/2016
132	06/05/2016
124	02/06/2016
117	15/05/2016
150	10/05/2016
133	07/05/2016