

October/November 2020

COS3711

Advanced Programming

85 Marks

Duration 2 Hours

EXAMINERS:

FIRST: MR CL PILKINGTON

SECOND: MR K HALLAND

EXTERNAL: DR L MARSHALL (UNIVERSITY OF PRETORIA)

This paper consists of 6 pages.

Instructions

1. You may type your answers in a word processor (and then print to PDF for submission) or
handwrite your answers (and then scan to PDF).
2. Answer all questions.
3. The mark for each question is given in brackets next to each question.
4. Please answer questions in order of appearance.
5. Note that no pre-processor directives are required unless specifically asked for.

**Remember to complete the Honesty Declaration when submitting your answers. By
submitting your answers you are confirming that this is your own, unaided work.**

The questions in this paper are based on a simplified scenario where records of patients admitted to a hospital with infections are kept.

A main client would be responsible for gathering all the required information from the user, where the recording is based on a central container or list of patient records, which is written to file when the application is closed and read in from file when the application is started. Patients (who have a patient number) are considered to be of two types: health workers (where it needs to be known where the patient works) and non-health worker patients (in which case the testing site needs to be known).

Each patient record also has an object indicating the infection that the patient has (assuming that each patient has only one infection). An infection (which has an infection code) could be viral (with a flag indicating whether there is an antiviral that is effective against it) or bacterial (with an indication of an antibiotic which is effective against it).

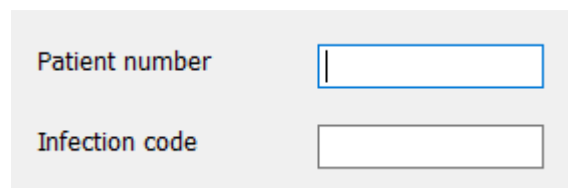
Question 1

[37 marks]

- 1.1 Create a UML class diagram to express the scenario described above, taking design patterns into consideration in the design.

You should include the necessary classes, class attributes, class operations, and class relationships to make it clear you understand how data will be managed and passed between classes. Include the Client class to be used. (20)

- 1.2 Assume a GUI with the following line edits is used to get some of the required information from users.



The image shows a light gray rectangular box containing two text input fields. The top field is preceded by the label 'Patient number' in a blue font. The bottom field is preceded by the label 'Infection code' in a blue font. Both fields are empty and have a thin blue border.

Consider the code used to set up the line edits and get the data from the user.

```
QLineEdit *inputPatientNumber = new QLineEdit();
QLineEdit *inputInfectionCode = new QLineEdit();
...
QString patientNumber = inputPatientNumber->text();
QString infectionCode = inputInfectionCode->text();
```

- 1.2.1 Which anti-pattern could be identified here? Explain your choice. (2)
- 1.2.2 Provide the code (based on that provided above) that uses a `QRegExpValidator` to force a user to enter a patient number in exactly the following format: (11)
- First 3 digits: number between 100 and 199.
 - Optional gender character (using an `f` or an `m`).
 - 3 or more alphabetic characters, where at least the first character is a capital letter.
- 1.2.3 Assuming that the validator in 1.2.2 has been correctly set up, can `200mSipho` be entered into `inputPatientNumber`? Explain why or why not. (1)

[TURN OVER]

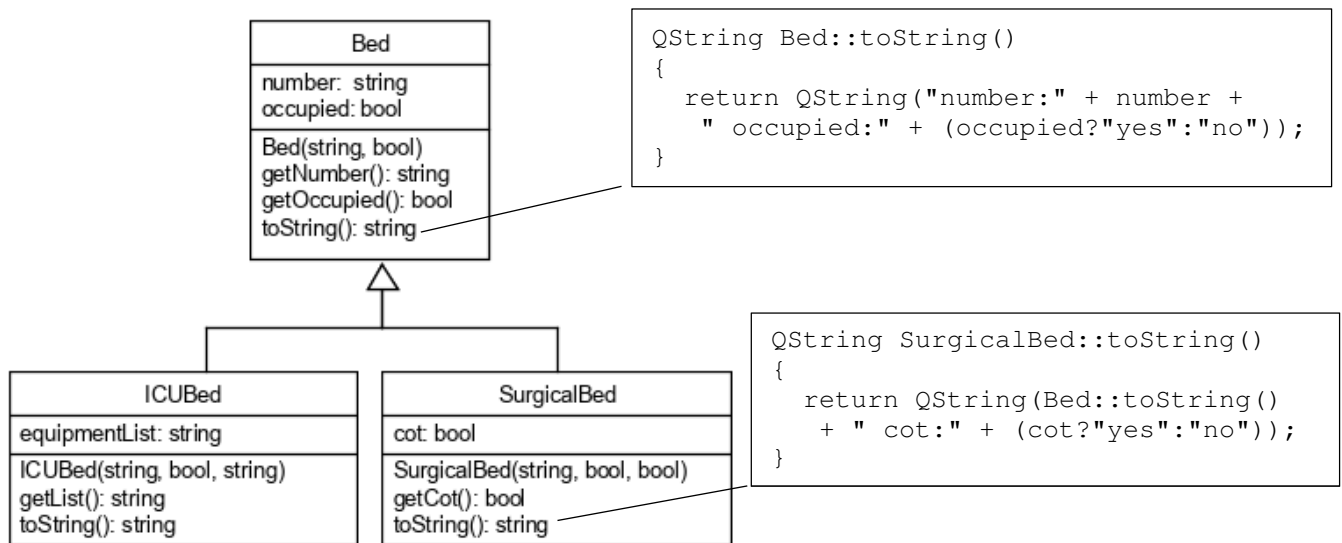
1.2.4 Assuming that the validator in 1.2.2 has been correctly set up, can 100S be entered into `inputPatientNumber`? Explain why or why not. (1)

1.2.5 It has been suggested that an input mask would work better than a regular expression validator in this scenario. Explain why you agree or disagree. (2)

Question 2

[19 marks]

The following UML class diagram and code fragments represents a partial view of the beds available at the hospital.



```

class Bed : public QObject
{
    Q_OBJECT
    Q_PROPERTY(QString number READ getNumber)
    Q_PROPERTY(bool occupied READ getOccupied)
public:
    Bed(QString, bool);
    QString getNumber() const;
    bool getOccupied() const;
    QString toString();
private:
    QString number;
    bool occupied;
};

class ICUBed : public Bed
{
    Q_OBJECT
    Q_PROPERTY(QString equipmentList READ getList)
public:
    ICUBed(QString, bool, QString);
    QString getList() const;
    QString toString();
private:
    QString equipmentList;
};
  
```

[TURN OVER]

```
class SurgicalBed : public Bed
{
    Q_OBJECT
    Q_PROPERTY(bool cot READ getCot)
public:
    SurgicalBed(QString, bool, bool);
    bool getCot() const;
    QString toString();
private:
    bool cot;
};
```

2.1 Consider the following code.

```
Bed sb = new SurgicalBed("000", true, false);
QString output = sb->toString();
```

The string `output` contains the following: `number:000 occupied:yes`

Explain why not all three pieces of data (`number`, `occupied`, and `cot`) are contained in the `output` variable and which OOP principles are involved. (4)

2.2 The following example XML output is required

```
<bed type="ICUBed">
  <number>999</number>
  <occupied>true</occupied>
  <equipmentList>HRT, ventilator</equipmentList>
</bed>
<bed type="SurgicalBed">
  <number>998</number>
  <occupied>true</occupied>
  <cot>false</cot>
</bed>
```

where the example objects used are

```
ICUBed icu = new ICUBed("999", true, "HRT, ventilator");
SurgicalBed surgical = new SurgicalBed("998", true, false);
```

The code below provides the skeleton code to produce the XML using DOM. You are required to add code into this skeleton to show how meta-objects will be used to gain read access to the properties of a `Bed` instance.

[If you are typing this answer, you can copy this code into your answer and just insert the correct code as needed.]

1	<code>QString xml = writeDom(icu).toString();</code>
2	
3	<code>QDomDocument writeDOM(Bed *object)</code>
4	<code>{</code>
5	<code> QDomDocument doc;</code>
6	<code> QDomElement bedElement = doc.createElement("bed");</code>
7	<code> bedElement.setAttribute();</code>

[TURN OVER]

```

8      doc.appendChild(bedElement);
9      QDomElement element = doc.createElement();
10     bedElement.appendChild(element);
11     QDomText text = doc.createTextNode();
12     element.appendChild(text);
13     return doc;
14 }

```

(11)

2.3 What impact would it make if the function `Bed::getNumber()` were made private? Consider access to the data in the object for both the cases where the meta-object is being used and where it is not. (2)

2.4 Is the following code legal? Explain your answer in detail.

```

Bed bed;
bed.setProperty("NOTE", "Isolation");

```

(2)

Question 3

[29 marks]

A design decision has been made to use Qt's model-view architecture to store the bed data. Two suggestions for this have been provided: using a `QAbstractTableModel` or a `QStandardItemModel`.

- 3.1 Explain in detail what the benefits and drawbacks are of using `QAbstractTableModel` for this scenario. (3)
- 3.2 Were a `QAbstractTableModel` to be used, how would the data be suitably visualised/seen by the user? (2)
- 3.3 Remembering that there are at least two types of beds that need to be managed by the model (as there may be empty cells in the table), can this be done in a `QStandardItemModel`, and if so, explain how? (3)

Bed Number	Occupied?	Equipment list	Cot?
998	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
999	<input checked="" type="checkbox"/>	HRT, ventilator	

- 3.4 As shown in the example display in 3.3, instead of displaying strings for "Yes" or "No" in columns where Boolean values are used, it is possible to use check boxes in these columns. Explain briefly how this can be done. (2)
- 3.5 It could be argued that a model to manage all hospital beds should be a singleton. What would you consider to be a convincing argument in this scenario? (2)
- 3.6 The following code has been provided to ensure the singleton design pattern. Does this meet the requirements of the classic singleton design pattern (in which case you should explain why), or not (in which case you should rewrite it to make it implement the pattern)?

[TURN OVER]

```

class BedsSingleton
{
public:
    QStandardItemModel* getModel();
private:
    BedsSingleton();
    static QStandardItemModel beds;
};

```

(4)

- 3.7 A print-to-file program, `printBedsToFile(QString outputRequired, QString filename)`, receives a `QString` parameter representing the type of list to print (such as all vacant beds, or a full list of all equipment linked to each ICU bed). It scans the bed data model and prints the output to the file given by `filename`.

This print program needs to be run as a separate process. Write the code necessary to start the print program as a separate process (passing "vacant" as the first parameter and "vacantbeds.txt" as the second. Use the `start()` function declaration below to guide you.

```

void QProcess::start(const QString &program,
    const QStringList &arguments,
    QIODevice::OpenMode mode = ReadWrite)

```

You should include instance creation where necessary.

(5)

- 3.8 Assume a `BedList` class.

BedList
list: QList<Bed*>
BedList() addBed(Bed*): void createMemento(): BedMemento* setMemento(BedMemento*): void

The `createMemento()` function code is as follows.

```

BedMemento* BedList::createMemento()
{
    QMultiMap<QString, Bed*> memento;

    for (int i=0; i<list->size(); i++)
    {
        const QMetaObject* meta = list->at(i)->metaObject();
        QString name = meta->className();
        memento.insert(name, list->at(i));
    }
    BedMemento* m = new BedMemento();
    m->setState(memento);
    return m;
}

```

Using this code as a guideline, give the class declaration for the `BedMemento` class so that it follows the classic Memento design pattern.

(8)