COS3711

May/June 2017

ADVANCED PROGRAMMING

Duration 2 Hours

80 Marks

EXAMINATION PANEL AS APPOINTED BY THE DEPARTMENT

Closed book examination

This examination question paper remains the property of the University of South Africa and may not be removed from the examination venue

EXAMINATION PANEL:	
FIRST:	MR CL PILKINGTON
SECOND:	MR K HALLAND
EXTERNAL:	DR L MARSHALL (UNIVERSITY OF PRETORIA)

Instructions

1 Answer all questions
2 Please write the numbers from 1 to 4 in the appropriate question number column on the examination answer book cover
3 All rough work must be done in your answer book
4. The mark for each question is given in brackets next to each question
5. Please answer questions in order of appearance
6 Note that no pre-processor directives are required unless specifically asked for

This examination paper consists of 6 pages

Good luck!

All the questions in this paper are based on the software requirements surrounding managing cell phones, their service providers and associated contracts.

Question 1 **[25 marks]**

A client application needs to manage a list of cell phone records where each record consists of the make of phone and its weight Also required is the functionality to read in records from an external file.

1 1 Provide a UML class diagram that can be used to design such a client application. This diagram should include the client, and show its relationship to other required classes Indicate data members and member functions as necessary to make the purpose of each class in the class diagram clear (12)

1 2 Given the definition of the `ServiceProvider` class below, answer the questions that follow.

```
class ServiceProvider
{
    Q_OBJECT
    Q_PROPERTY(QString name READ getName WRITE setName)
public
    ServiceProvider();
    ServiceProvider(QString n, QString ws),
private:
    QString name;
    QString website;

    void setName(QString n),
    void setWebsite(QString ws),
    QString getName() const;
    QString getWebsite()const;
};
```

1 2 1 This class definition results in a build error. Correct the error You do not need to rewrite the whole class definition, you only need to indicate clearly where you make the correction. (1)

1 2 2 What is the purpose of the macros present in the class definition? Briefly explain the practical programming benefits of a class that implements them (2)

1 3 Assuming that the build error in the class definition given in 1 2 has been corrected, write down what will be returned by the following function when the code below is executed (3)

```
QString function(const ServiceProvider* sp)
{
    QStringList allInfo,
    const QMetaObject *m = sp->metaObject();
```

```
        for(int i =meta->propertyOffset(); i <meta->propertyCount(); i++)
        {
            const QMetaProperty prop = meta->property(i);
            allInfo.append(QString(prop.name()));
            allInfo.append(prop.read(b).toString());
        }
        return allInfo.join(",");
    }

    ServiceProvider* sp =
                new ServiceProvider("ProviderA", "www.providerA.co.za");
    qDebug() << function(sp);
```

1.4 Assuming that the build error in the class definition given in 1.2 has been corrected, is the following code legal? If so, explain its effect; if not, explain why not. (2)

```
ServiceProvider* sp = new ServiceProvider();
sp->setProperty("website", "www.sp.co.za");
```

1.5 Consider the following code fragment

```
Phone* p = new Phone();
```

1.5.1 Write the code necessary to create a `QSharedPointer` for `p`. (2)

1.5.2 Describe what a `QSharedPointer` is, and how it is of benefit to a programmer (3)

Question 2 **[12 marks]**

It has been decided to use a model-view approach to manage the various cell phone packages that are available. Two different item-based solutions have been proposed.

a) `QTableWidget *model = new QTableWidget();`
b) `QStandardItemModel *model = new QStandardItemModel();`

2.1 What does it mean that these are item-based solutions? (1)

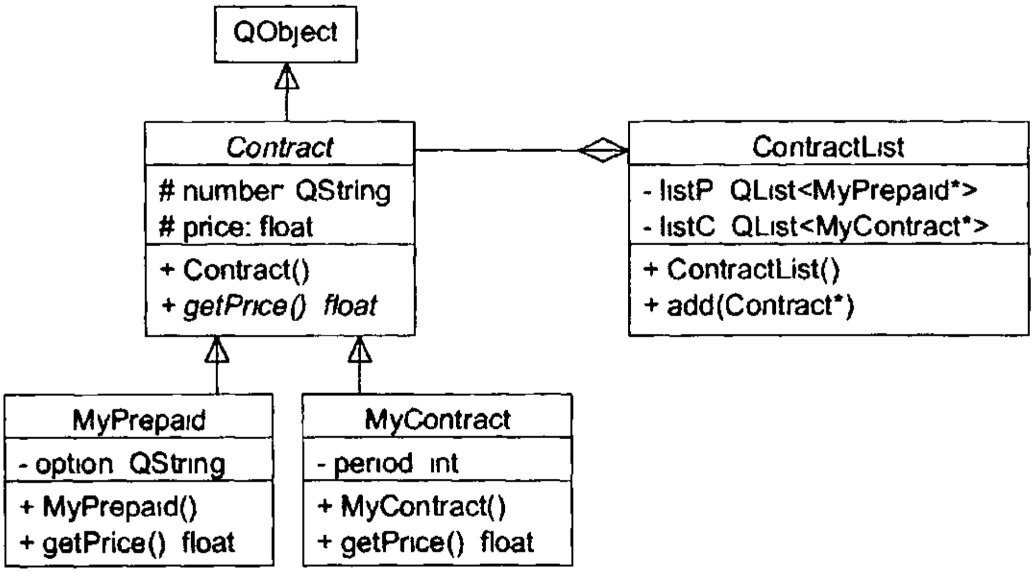
2.2 Discuss the fundamental difference between the two approaches proposed. Be sure to indicate which model you are discussing when making the comparison. (4)

2.3 Which of the two approaches is the less flexible, and in what two ways is it so? (3)

2.4 How is the MVC pattern an application of the Observer pattern, and what mechanism does Qt use to implement the necessary inter-object communication? (4)

Question 3 **[16 marks]**

Consider the UML class diagram below representing `Contract` types and a list of `Contracts`



3 1 Write the class definition for the `Contract` class, you are not required to write any implementation code (4)

3 2 Extend the UML class diagram by adding a `ContractFactory` class that implements a Factory Method design pattern for the concrete `Contract` types Include the necessary functions in the `ContractFactory` UML class diagram and uses/creates relationships; however, for the remaining classes, you need only include the class name (4)

3 3 Assuming that the user choice for a specific contract type is stored in the variable `QString choice`, write code that will return the required object using the Factory Method pattern from question 3 2 above. The `choice` variable is the only object that you may assume exists. (3)

3.4 The `ContractList` class has separate `QLists` for the two concrete types, however, the `ContractList add(Contract*)` function accepts only a pointer to the abstract base class Write code for the `add()` function so that the object pointer passed is added to the correct list (without having to add `getType()` type functions to the `Contract` hierarchy) (5)

Question 4 [27 marks]

Consıder the partıal class below that wıll manage runnıng processes

```
class Manage   publıc QDıalog
{
    Q_OBJECT

publıc
    Manage(QWıdget *parent = 0),
    ~Manage(),
    voıd runProcessContracts(),
prıvate
    QTextEdıt* output,
},
```

A program named `processContracts exe` ıs used to carry out operatıons on contract detaıls and return data ın the form `number#message`

4 1 Wrıte code that would be placed ın the functıon `runProcessContracts()` that wıll run `processContracts exe` as a separate process, assumıng that all requıred varıables stıll have to be declared (3)

4 2 Assumıng that the data ıs returned from thıs new process vıa the standard output, wrıte code that
a) wıll lısten for thıs output ın `runProcessContracts()`, and
b) show how the data wıll be read ın and wrıtten to the `output QTextEdıt` by addıng the necessary functıons to the class header, and ımplementıng such functıons.
HINT· Consıder usıng ınter-object communıcatıon. (8)

4 3 An `XMLWriter` class has been wrıtten wıth a `wrıte(QStringLıst strlıst)` functıon that wrıtes the data passed to ıt to an XML fıle usıng `QXMLStreamWrıter` (where `writer` ıs an ınstance of `QXmlStreamWrıter`). The strıng lıst passed to the functıon ıs made up of a lıst of `number#message` strıngs See the partıal functıon below

```
void XMLWrıter: wrıte(QStrıngLıst strlıst)
{
    QFıle fıle("messages xml");
    fıle open(QIODevıce:·WrıteOnly),

    wrıter.setDevıce(&fıle),
    wrıter setAutoFormattıng(true),
    wrıter wrıteStartDocument();
    wrıter.wrıteStartElement("messages"),

    for (ınt ı=0; ı<strlıst->sıze(), ı++)
    {
        // splıt the strıng at the # character
        // create the <contract> element wıth ıts attrıbute
        // create the <message> element wıth ıts text
        // remember to close off the <contract> elements
    }

    wrıter wrıteEndElement(), // end of messages element
    wrıter wrıteEndDocument(),
    file close(),
}
```

Write the code in the `for` loop to produce XML in the following format (using the string `0999999999#The message` as an example)

```
<contract number="0999999999">
  <message>The message</message>
</contract>
```

(8)

4.4 It has been decided to run an `XMLReader` task, that reads an XML file, as a thread. The following code has been provided for the header file

```
class XMLWriter . public QThread
{
  public
    XMLWriter(),
    ~XMLWriter(),

    void run(); // this function does the reading task

  signals:
    void returnItem(QString), // signal to return each number#message string
},
```

Rewrite this code so that it uses the recommend approach to threading without adding any extra functions. It is not necessary to include the comments (3)

4.5 Writing to, and reading from, XML as used above is an example of which design pattern? Discuss in detail how this pattern differs from the Memento design pattern, pointing out two major differences (5)