

INF2611 May/June 2017 Examination Memorandum

Question 1: Menus and Widgets (12)

The entries in the menu bar can be created in two ways:

- 1.1 By creating an action in the Action Editor and dragging and dropping it into a menu. Each action dropped into the menu will act as an individual menu entry.
- 1.2 By typing text for menus and menu entries in the menu bar replacing the Type Here placeholders. In that case, each menu entry will appear as an individual action in the Action Editor, where you can configure its properties.

1.3 Dock widget

- a) A Dock widget is created with the `QDockWidget` class.
- b) A Dock widget can be used to create detachable tool palettes or widget panels.
- c) Allowable dock areas are `LeftDockWidgetArea`, `RightDockWidgetArea`, `TopDockWidgetArea`, and `BottomDockWidgetArea`.
- d) True

Question 2: Database handling (18)

2.1 A traditional file system lacks several features such as:

- Indexing
- Encryption
- Joining or merging files

A traditional file system is also not as efficient in handling large volumes of data.

A DBMS has all these features and also has features such as:

- Auto backup
- Data Sharing
- SecurityIntegrity

2.2 MySQL data types:

- `smallint` - integer value
- `char` - Fixed length strings up to 255 characters
- `varchar` - Variable length strings up to 255 characters
- `longblob` - Large blocks of binary data

- 2.3.1** The four parameters required for the connect() method are:
1. host name - specifies the location of the MySQL database server
 2. user name - authorised user name to access the database
 3. password - the password for the authorised user
 4. database name - the name of the database to connect to

- 2.3.2** The remaining three database-related methods in the code are:
1. cursor() - Returns the cursor object from the connection. The cursor object is used to traverse the records from the result set
 2. execute() - Used to execute the SQL statement
 3. close() - Disconnects the database connection

Question 3: Console-based database handling (25)

3.1

```
INSERT INTO books (book_isbn, book_name, quantity, price)
VALUES (978409506, 'Python for you', 1, 100
"")
print('One row inserted into the products table')
```

3.2

```
try:
    cursor.execute ("SELECT * from book")
    print ("Book ISBN\tBook Name\tQuantity\tPrice")

    while(1):
        row=cursor.fetchone()
        if row==None:
            break
        print ("%d\t\t%s\t\t%d\t\t%f" %(row[0], row[1], row[2],row[3]))
except MySQLdb.Error:
    print ("Error in fetching rows")
```

3.3

```
p=int(input("Enter Product ID: "))
cursor.execute ("SELECT * from products where prod_id=%d" %p)

row=cursor.fetchone()

if row==None:
    print ("Sorry no Product found with ID %d" %p)
else:
    print ("Information of the product with ID %d is as
follows:" %p)
    print ("Product ID: %d, Product Name: %s, Quantity: %d,
Price: %f" %(row[0], row[1], row[2], row[3]))
```

Question 4: GUI-based database handling (15)

4.1 QSqlDatabase

4.2 Four methods of the class used to intergrate databases with PyQt:

1. `addDatabase()` - Used to specify the database driver of the database to which you want to establish connection. It is through the database drivers that the database is accessed.
2. `setHostName()` - Used to specify the hostname.
3. `setDatabaseName()` - Used to specify the name of the database that you want to work with.
4. `setUserName()` - Used to specify the name of the authorized user through whom you want to access the database.
5. `setPassword()` - Used to specify the password of the authorized user to access the database.
6. `open()` - Opens the database connection using the current connection attributes. The method returns a Boolean true or false value, depending on whether the connection to the database is successfully established or not.
7. `lastError()` - Used to display error information.

4.3 To display the rows fetched from the database table, you will use a Table View widget. The Table View widget will display database table information in tabular format. When fetching and displaying information from a database, you want to use a model that is easy to deal with. A model is a mirror image of the database table that the user can use to navigate and edit if required. To create a model, you need to create an instance of the `QSqlTableModel` class.

The `QTableView` class is used to create a Table view that displays items from a model. That is, information can be displayed in the Table view through the model.

4.4 The two methods of the `QSqlTableModel` class that are used to display rows that are fetched from a database table:

1. `setTable()` - Used to specify the database table you want the model to work with.
2. `setEditStrategy()` - Applies the strategy for editing the database table.
3. `select()` - Used to populate the model with the information of the database table specified with `setTable()`

Additional Questions

Question 1 (Advanced Widgets)

1. What are widgets and why are they useful in programming? (2)
2. Name any 3 widgets that you have used in programming (3)
3. Explain how you would add a LCD widget and how you can use this to show the system time (5)
4. Explain how you would add a calendar widget and how you can use this to display the date selected in a date edit widget.
5. Why is a combobox useful in applications? (1)
6. Explain how can you use a combobox in a GUI (3)

Answer

1. The main component for creating a user interface is widgets. Input widgets are used for interacting with the user. Display widgets are used for displaying information or messages to the user. It allows for the creation of an interface quickly and easily.
2. Buttons, Menus and scrollbars
3. Adding an LCD Widget:
 - a. Firstly, go to Qt and drag and drop a LCD number Widget Then convert the QT program to python code.
 - b. Create python code to connect the signal to a function.
 - c. Create an instance of QTimer and set its timeout to 1000ms.
 - d. Whenever the timer is generated it invokes the function.
 - e. The functions will fetch the system clock, convert it to a string and make it display in the LCDNumber Widget.
4. Displaying the date in a date edit widget when selected on a calendar:
 - a. Firstly, go to Qt and drag and drop a Calendar and Date Edit Widget Then convert the QT program to python code.
 - b. Create python code to connect the signal to a function.
 - c. Whenever the date is changed by clicking in a new date the function is invoked.
 - d. The date selected by the user is retrieved by selectedDate() and displayed in the date edit widget with the setDate() method.

5. A comboBox can be used to trigger certain actions and is usefull for saving space on the screen:
 - a. Firstly, go to Qt and drag and drop a Calendar and Date Edit Widget Then convert the QT program to python code.
 - b. Create python code to connect the signal to a function.
 - c. You can add code to add items into the comboBox so that when selected, together with other widgets like a push button, will generate a result. A function then operates depending on what is true in the programs IF statement.

Menus & Toolbars

1. Explain how would you create a Menu bar with File and under File three placeholders with Edit, copy and paste (2)
2. Explain how you would you create a shortcut key for the copy and paste submenus (4)
3. How is a menu bar different from a toolbar (2)
4. What is a placeholder? (1)
5. How would you add a separator in a menu to create a nested menu? (2)
6. Explain how does an Action Editor works (4)

Creating a Toolbar

1. What is a toolbar useful for (1)
2. Explain briefly how you would create an action with an icon

Dock Widget

1. What is a docked widget and why is it useful?
2. Name 4 areas that you can place a dock widget.
3. What property should you enable to make the dock area movable?

Tab Widget

1. What is the purpose of a tab widget?
2. How do you convert a tab to a widget?
3. How do you change the style, gradient and colour of a tabbed document?

Other Questions on Menus

1. Give 4 reasons why it is important to use Menus in an application (3)
2. Give the necessary steps on how to add a menu to an item (2)
3. Explain how to attach code to menu items? Give an example by referring to an application (3)
4. Explain the different ways that can be used to trigger the same operation. (3)
5. Discuss 3 reasons why menus are required in the development of a database application (2)

Question 3

1. Describe the characteristics of a Multiple Document Interface (MDI) – (5)
2. Discuss 3 Reasons/benefits for the use of multiple forms in an application AND give motivation with reference to the development of the project application. (3)
3. Identify 2 form / window modes.

Answer

1. Characteristics of MDI
 1. Consists of a main window which had a menu bar, toolbar and central workspace
 2. One acts as parent and the others a child
 3. Central workspace manages the child widgets
 4. It is represented by the MdiArea
 5. Child windows are also called subwindows.

2. Benefits for the use of multiple forms:
 1. With multiple document interfaces (and also tabbed document interfaces), a single menu bar and/or toolbar is shared between all child windows, reducing clutter and increasing efficient use of screen space. This argument is less relevant on an operating system which uses a common menu bar.
 2. An application's child windows can be hidden/shown/minimized/maximized as a whole.
 3. Features such as "Tile" and "Cascade" can be implemented for the child windows.
 3. Modal and Modeless

Question 4

1. What are the benefits of storing information in a database? (3)
2. What is the difference between data and information?
3. What are the benefits of using MySQL (6)
4. Give the commands you would use to create a database called shopping using MySQL:
 - a. What is the command used to create a database called shopping (1)
 - b. How would you display any tables in the shopping database (1)
 - c. Suppose you have created a table called products. How would you display the contents of products. (1)

Answer

1. The benefits of storing information in a database are:
 - a. Multiple users can open, save and update information simultaneously.
 - b. Databases can be maintained with backups and optimisation.
 - c. Databases are more reliable and can be setup for high performance and high availability.
2. Data is raw/unfiltered unstructured whereas information is structured/filtered data which is somehow useful to the user.
3. The benefits of using MySQL are:
 - a. MySQL is very popular amongst developers.
 - b. It is an open source system which means there is no need for licence.
 - c. It takes less storage and has remarkable performance.
 - d. Available for Windows, Unix, Linux and Mac OS.
 - e. It is easy to maintain and upgrade.
 - f. It has an efficient query engine
4. Commands used to create a table called shopping using MySQL:
 - a. Create table - `create database shopping;`
 - b. Display any tables in the shopping database - `show tables;`
 - c. Display the contents of the products table - `describe products;`

5. Use the code below to:
- Create a new table called cars with name, surname, date, address and age into the database. Provide the single line of code only.
 - Add the following values into the database. Mike, Praisley. 23-12-1995, 23 First road, 67. Provide the single line of code only.

```
#InsertRow.py
import sys

import mysql.connector

conn =
mysql.connector.connect(host="localhost",user="root",password="nowin
",database="booking")

cursor = conn.cursor()

cursor.execute(Insert code here to add the items)")

print ("One Row Added")

cursor.close()

conn.commit()

conn.close
```

Answer

5. Code to:
- Create a new table called cars with name, surname, date, address and age into the database.

```
#InsertRow.py
import sys
import mysql.connector
conn =
mysql.connector.connect(host="localhost",user="root",password="test"
,database=" shopping ")
cursor = conn.cursor()
cursor.execute("create table cars(Name Varchar, Surname Varchar,
DateValue DATE, Address Varchar)")
print ("One Row Added")
cursor.close()
conn.commit()
conn.close
```


- b. Code to add the following values into the database. Mike, Praisley. 23-12-1995, 23 First road, 67

```
#InsertRow.py
import sys
import mysql.connector
conn =
mysql.connector.connect(host="localhost",user="root",password="test"
, database="shopping")
cursor = conn.cursor()
cursor.execute("INSERT INTO products VALUES ('Mike', 'Praisley', 23-
12-1995, 23 First road)")
print ("One Row Added")
cursor.close()
conn.commit()
conn.close
```