

## Question 1 (Advanced Widgets)

### 1 What are widgets and why are they useful in programming? (2)

Widgets are items which facilitate the building of Graphical User Interfaces which specify a specific type of user-interaction. They are precoded which saves programmers a lot of time. They provide a quick and easy way for designing user interfaces.

### 2 Name any 3 widgets that you have used in programming (3)

ComboBox, LineEdit, Calendar Widget, LCD Widget

### 3 Explain how you would add a LCD widget and how you can use this to show the system time (5)

- Drag and Drop the LCD Number Widget onto a dialog
- Then convert the ui to a python file
- You create an instance of QTimer and assign it to a name e.g. timer.  
`timer=QtCore.QTimer(self).`
- You connect the timeout method to a function that fetches the system clock.  
`timer.timeout.connect(self.showlcd)`
- you then add the timing to the timeout function: `timer.timeout(1000)` every 1000ms
- Then for every 1000ms, you show the `self.showlcd()`
- The `showlcd` method connects to an instance of the systems clock. `time = QtCore.QTime.currentTime()`
- You then convert the time to text and display it via the LCDNumber.  
`text=time.toString('hh:mm")`                      `self.ui.LCDNumber.display(text)`

### 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.

- Drag and Drop the QCalendarWidget onto a dialog and a dateEditWidget
- Then convert the ui to a python file
- You then connect the calendars on `selectionChanged()` Signal to a slot method that will display the date from the calendar on the dateEditWidget.
- `QtCore.QObject.connect(self.ui.calendarWidget, QtCore.SIGNAL('selectionChanged()'), self.dispdate.`
- `def dispdate(self):`  
`self.ui.dateEdit.setDate(self.ui.calendarWidget.selectedDate())`

**5 Why is a comboBox useful in applications? (1)**

It displays the items that are listed using minimal screen space.

**6 Explain how can you use a combo box in a GUI (3)**

You drag a combo box into the dialog

You convert the file to a py file.

You then add items to the combo box by using the ComboBox.addItem() method

You can add the items of a combo box using a loop

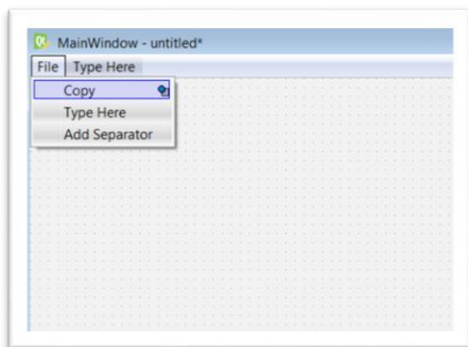
To change screens of a stackedView

To select items of text

Pixmap

**Question 2 (Menus and Toolbars)**

Use the diagram below to answer the questions that follow.



**7 Explain how would you create a Menu bar with File and under File three placeholders with Edit, copy and paste (2)**

Where it says type here you type here Type File and Press Enter. Then Type Edit and press Enter, Type Copy – press Enter, type Paste, Press Enter

**8 Explain how you would you create a shortcut key for the copy and paste submenus (4)**

You activate the action editor. Select on the action item for copy and double click. A window pops up and you type the shortcut key in the place allocated for shortcut. You do the same for paste.

**9 How is a menu bar different from a toolbar (2)**

A menubar can have many drop down menus with many items for each list. You can only have one toolbar and it uses icons.

**10 What is a placeholder? (1)**

A placeholder is a place that has been reserved so that the user can add text or a picture to it.

**11 How would you add a separator in a menu to create a nested menu? (2)**

Double click the add Separator option in the drop down menu

**12 Explain how an Action Editor works (4)**

An action editor initiates actions generated by the user. The action editor works with user initiated actions via menu bars, toolbars and shortcut keys. On occurrence of an action and function is executed in response to the action initiated.

**Creating a toolbar**

**13 What is a toolbar useful for (1)**

A toolbar displays icons instead of text to represent the task that it can perform

**14 Explain briefly how you would create an action with an icon**

You would go to the action editor, select new, Give the action a name, assign an icon to it either through the file system or a resource file. Then drag the action to a toolbar.

**DOCK WIDGET**

**Questions**

**15 What is a docked widget and why is it useful?**

**16 Name 4 areas that you can place a dock widget**

LeftDockWidgetArea  
RightDockWidgetArea  
TopDockWidgetArea  
BottomDockWidgetArea

**17 What property should you enable to make the dock area movable**

DockWidgetMovable

**TAB WIDGET**

**Questions**

**18 What is the purpose of a tab widget**

It is used to create detachable tool pallets and widget panels

**19 How do you convert a tab to a widget**

Right click on it and select Morph into

**20 How do you change the style, gradient and colour of a tabbed document?**

Through the style sheet

**21 Give 4 reasons why it is important to use Menus in an application (3) ????**

- users rely on **menus** to find content and **use** features.
- Menus can have multiple items listed under each menu which in turn can also have submenus, as well as separators. Each item in a menu performs an action related to the action Editor.

- 22 Give the necessary steps on how to add a menu to an item (2) ?? An item to a menu?**

On the placeholder which says Type Here, type in the name of the menu. To add an item, drop down the menu and enter in your text. Press enter every time.

Or

Go to the action editor. Create a new Action and drag and drop to the menu bar.

- 23 Explain how to attach code to menu items? Give an example by referring to an application. 3**

You use connect the action editor item via a signal clicked function to a slot which you write the method for. In this case. openMessage

```
QtCore.QObject.connect(self.ui.actionOpen, QtCore.SIGNAL('clicked()'), self.openMessage)
```

- 24 Explain the different ways that can be used to trigger the same operation. (3)???**

- 25 Discuss 3 reasons why menus are required in the development of a database application (2) ????**

Menus allow for the user to easily navigate to different areas within the database.

Menus can be actions to allow the user to Read, Update, Create nEW Records and Delete records from the data base.

### **Question 3 (Multiple Document Interface)**

- 26 Describe the characteristics of a Multiple Document Interface (MDI) – (5)**

Main Window containing a menu bar, tool bar and a central QWorkspace widget as well as other child windows. The Main Window is the parent and all dialogs and Widget within the main Window are children.

- 27 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)**

- 28 Identify 2 modes that you can use in a database project.**

- 29 Use the code below to fill in the code where specified.**

```
import sys
from mddemo import *

class MyForm(QtGui.QMainWindow):
    def __init__(self, parent=None):
        QtGui.QWidget.__init__(self, parent)
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.mdiArea.addSubWindow(self.ui.Main)
        self.ui.mdiArea.addSubWindow(self.ui.Tab1)
        self.ui.mdiArea.addSubWindow(self.ui.Tab2)
        self.ui.mdiArea.addSubWindow(self.ui.Tab3)
        self.ui.Main.setWindowTitle("Main")
        QtCore.QObject.connect(self.ui.button1, QtCore.SIGNAL('clicked()'), self.displayNext)
        QtCore.QObject.connect(self.ui.ShowPrevious, QtCore.SIGNAL('clicked()'),
self.displayprevious)
        QtCore.QObject.connect(self.ui.closeAll, QtCore.SIGNAL('clicked()'), self.closeAll)
        QtCore.QObject.connect(self.ui.cascadeButton, QtCore.SIGNAL('clicked()'),
self.cascadeArrange)
        QtCore.QObject.connect(self.ui.tileButton, QtCore.SIGNAL('clicked()'), self.tileArrange)
        QtCore.QObject.connect(self.ui.SubWindowViewButton, QtCore.SIGNAL('clicked()'),
self.SubWindowView)
        QtCore.QObject.connect(self.ui.TabbedViewButton, QtCore.SIGNAL('clicked()'),
self.TabbedView)
        QtCore.QObject.connect(self.ui.actionFirst_Window, QtCore.SIGNAL('triggered()'),
self.displayNext)
        QtCore.QObject.connect(self.ui.actionSecond_Window, QtCore.SIGNAL('triggered()'),
self.displayprevious)

    def displayNext(self):
        self.ui.Court2.setFocus()

    def displayprevious(self):
        self.ui.mdiArea.activatePreviousSubWindow()
```

```

def closeAll(self):
    self.ui.mdiArea.closeAllSubWindows()

def cascadeArrange(self):
    self.ui.mdiArea.cascadeSubWindows()
    Enter code to cascade all subwindows

def tileArrange(self):
    self.ui.mdiArea.tileSubWindows()
    Enter Code to arrage windows in a tile fashion

def SubWindowView(self):
    self.ui.mdiArea.setViewMode(0)
    Enter code to set view to Window View

def TabbedView(self):
    self.ui.mdiArea.setViewMode(1)
    Enter code to set view to tabbed view

if __name__ == "__main__":
    app = QtGui.QApplication(sys.argv)
    myapp = MyForm()
    myapp.show()
    sys.exit(app.exec_())

```

#### Question 4 (Database handling)

##### 30 What are the benefits of storing information in a database? (3)

- Information can be encrypted/decrypted
- Backups can be performed
- Restoration of data can be performed
- Administration can be managed.
- It handles large volumes of data and can retrieve data speedily by using Indexes
- Data sharing
- Integrity

##### 31 What is the difference between data and information?

Data is a series of facts. Information is when that data is turned into information that can be used to make decisions

##### 32 What are the benefits of using MySql (6)

It is open source and can be used on multiple platforms like Linux, Windows, iOS and Windows

It provides encryption and decryption  
It is popular amongst web developers  
Takes less storage space  
Is free – open source  
Is easy maintain and upgrade  
It has a very efficient query engine.

**33 Give the commands you would use to create a database called shopping using MySQL.**  
create database shopping;

**34 How would you display any tables in the shopping database (1)**  
show tables;

**35 Suppose you have created a table called products. How would you display the contents of products.**  
select \* from products;

**36 Use the code below too**

- a. Create a new table called cars with name, surname, date, address and age into the database. Provide the single line of code only  
create table cars (name char(25), surname char(25), date date, address tinytext, age smallint)
- b. 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 into cars (name, surname, date, address, age)
values ('Mike', 'Praisley', 23-12-1995, '23 first road, 67')insert code here to add the
items)")

print ("One Row Added")

cursor.close()
conn.commit()
conn.close
```

2. Use the program below to comment on the code given. You do not need to type out the entire code, only comment. (8)

```

import sys from
assignment2 import *
from PyQt4 import QSql, QtGui

def createConnection(): db =
QSql.QSqlDatabase.addDatabase('QMYSQL')
db.setHostName('localhost')
db.setDatabaseName('booking')
db.setUserName('root')
db.setPassword('nowin')
db.open()
print (db.lastError().text())
return True

class MyForm(QtGui.QMainWindow):
def __init__(self, parent=None):
    QtGui.QWidget.__init__(self, parent)
self.ui = Ui_MainWindow()
self.ui.setupUi(self)
#Comment here Constructor for the main window
self.model = QSql.QSqlTableModel(self)
#Comment here sets the model to an instance of the TableModel for SQL to display a database via a table and also for editing
self.model.setTable("players2")
#Comment here The table that must be set to the instance of the Table model is players 2
self.model.setEditStrategy(QSql.QSqlTableModel.OnManualSubmit)
#Comment here The edit strategy is set on Manual Submit which means that all the changes are cached until submitAll()
method is called
self.model.select()
#Comment here Activate the Model by selecting it..?????

self.ui.tableView.setModel(self.model)
#connects the button UpdateButton to the function UpdateRecords.
QtCore.QObject.connect(self.ui.UpdateButton, QtCore.SIGNAL('clicked()'),
self.UpdateRecords)
QtCore.QObject.connect(self.ui.CancelButton, QtCore.SIGNAL('clicked()'),
self.CancelChanges)
QtCore.QObject.connect(self.ui.DeleteButton, QtCore.SIGNAL('clicked()'),
self.DeleteRecords)

def UpdateRecords(self):
#submits the current edited row
self.model.submitAll()

def CancelChanges(self):
#cancels and changes
self.model.revertAll()

def DeleteRecords(self):
#deletes a row
self.model.removeRow(self.ui.tableView.currentIndex().row())
self.model.submitAll()

if __name__ == "__main__":
app = QtGui.QApplication(sys.argv)
if not createConnection():
sys.exit(1)
myapp = MyForm()
myapp.show()
sys.exit(app.exec_())

```



## CHAPTER 7: CREATING A GUI

### 1 **What is the difference between mode and modeless.**

A modal dialog is one that blocks the user from interacting with other parts of the application. The object that is modal is the only object that the user can interact with. Modeless dialog is where the user can interact with the rest of the application whilst the current dialog is active.

### 2 **What is the base class for all user Interface classes?**

QWidget

### 3 **Explain Signals and Slots**

A signals and slots are event handlers. A signal is an event and a slot is a method that executes when the signal is emitted.

### 4 **What is the sys.argv statement.**

It is when the system arguments that get fetched from the command line and passed to the method while creating the application. It helps in passing and controlling the startup attributes of script.

### 5 **What is the purpose of sys.exit**

It ensures a clean exit and releases any memory resources.

### 6 **What is the super class of a Widget.**

QWidgetClass

### 7 **Name three top level Widgets.**

QDialog  
QMainWindow  
QWidget

### 8 **What does Tristate mean?**

The user has the option of neither checking or unchecking a check box.

## CHAPTER 9: ADVANCED WIDGETS

### 9 What is the syntax to set the current scene to Scene?

`QGraphicsView.setScene(self, QGraphicsScene.scene)`

### 10 When signal is omitted when the Graphics scene is being viewed?

`QGraphicsScene.changed()`

### 11 What is the difference between a Graphics Scene and Graphics View

Graphics Scene is a container that holds all the Graphics. Graphics View is what enables the Ui to display Graphics.

## CHAPTER 10: MENUS AND TOOLBARS

### 12 How do you remove the menu bar from an instance of QMainWindow class?

Right clicking on the Main Window, brings up the context menu where you can remove the menu bar.

### 13 How do you add a menu bar after you've removed it?

Right click on the main window and select Create menu bar from the context menu.

### 14 How do you add menu items to a menu.

You type in the type here Placeholder and press enter when you are done.

You add an Action to the action editor and then assign it to a menu by dragging and dropping it on the menu

### 15 Which two views does the action editor have?

classic View

Detailed View

### 16 What four states can an action be in?

Normal

Disabled

Active

Selected

### 17 For a menu item, where can you set a statusTip?

in the property window. A tip is provided when the user hovers over the menu

### 18 When placing items on a toolbar, from which two places can you select icons from?

disk drive

resource file

### 19 What 6 things appear in the action editor?

name of the action

checkable?

Shortcut Key

ToolTip

used?

Text

**20 How can actions be initiated by the user on a user interface.**

through the menu, through a toolbar or via a shortcut key.

**21 What is the extension for a resource file?**

qrc

**22 How do you create a resource file?**

Select the resource Browser, Click on the Edit Resource icon. The resource browser will prompt you for a name for the resource file. To add a resource, you need to add a prefix to the resource file. The you need to add a prefix, which is a section or category within the resource file. Add Prefix by selecting the add prefix icon.

**23 What is a prefix?**

A section or category in a resource file to categorise resources.

## CHAPTER 11: MULTIPLE DOCUMENTS AND LAYOUTS

**24 What class is used to create Dock Widget**

QDockWidget Class

**25 What is the AllDockWidgetFeatures do when selected in the property editor?**

It allows the widget to be moved to any area, to be detached from the main window and floatable as an independent Widget. on top of the other widnows, it is closable.

**26 What does a MDI consist of?**

Main Window containing a menu bar, tool bar and a central QWorkspace widget

**27 What can a Dock widget be used for?**

Used to create detachable tool palettes or widget panels.

**28 Name four dock areas?**

LeftDockWidgetArea

RightDockWidgetArea

TopDockWidgetArea – below the toolbar

BottomDockWidgetArea

**29 If selected, What does the all Dock Widget Feature allow:**

DockWidgetClosable (DockWidget can be closed, DocWidget can be moveable, DockWidget can be floatable.

**30 Provide 4 Properties of a dock widget**

DockWidgetClosable

DockWidgetMoveable

DockWidgetFloatable

## CHAPTER 12: DATABASES

### **31 Why use MySQL?**

- 31.1 Open Source, it is available on several platforms, eg. Windows, Unix, Linux, Free BSD, Mac OS
- 31.2 Good security. Include encryption/decryption and other security methods.
- 31.3 Efficient Query Engine.
- 31.4 Easy to upgrade and maintain
- 31.5 Takes less storage space

### **32 What are the benefits of storing information in a database:**

- 32.1 fetching data is faster and more efficient than traditional filing systems.
- 32.2 Databases use indexes, hashing and other schemes to quickly find data
- 32.3 Databases have autobackup and restore facilities
- 32.4 encryption for high security
- 32.5 They have built in integrity constraints

### **33 What method do you use to ensure that any changes you made to the database are applied?**

- 33.1 `commit()`

### **34 What data type is tinytext?**

- 34.1 The smallest of long blocks of text data

### **35 What is the cursor() method used for in a database?**

- 35.1 used to traverse through the result set. It is almost like a place holder.

### **36 What is the execute() method used for.**

- 36.1 It is used together with the connection to execute the SQL statement

### **37 Which command displays all the tables in a database?**

- 37.1 `show tables;`

### **38 Which command is used to show the layout of a table?**

- 38.1 `describe tableName;`

### **39 Name 3 methods of the QSqlDatabase Class**

- 39.1 `addDatabase()` - used to add the drivers of a particular database system
- 39.2 `open()` - opens the connection to a particular database
- 39.3 `setDatabaseName` - sets the name of a particular database
- 39.4 `setHostName` - sets the host name
- 39.5 `lastError()` - used to display error info that might occur.

**40 What is the QSqlTableModel class used for?**

40.1.1 To display a database table in tabular format

**41 When using the Qt Gui to access a database via QSqlTableModel what methods are applied for Edit strategies can a table have and what is the difference between them (10)**

OnFieldChange	All changes applied to the model will be applied immediately to the database table
OnRowSubmit	All modification made to a row will be applied when the user moves to another row
OnManualSubmit	All modifications will be cached and implemented via a the SubmitAll method.
RevertAll	All modifications for OnManualSubmit that were cached will be undone.
SubmitAll	All modifications for OnManualSubmit that were cached will be applied.

**42 Say you wanted to set the Table Model to work with the Products table. What method would you use?**

42.1 `setTable("products")`

**43 Why do you need to import QSql module into the program when maintaining databases through PyQt**

You need to import the drivers and classes to integrate the database into PyQt applications.

**44 Write a function called createConnection in order to code to create connection to a database called megastores. The password : 123, host: localhost, user: root. Remember to open the connection, create a method for printing the last error.**

```
def createConnection():
    db = QSql.QSqlDatabase.addDatabase('QMYSQL')
    db.setHostName('localhost')
    db.setDatabaseName('megastores')
    db.setUser('user')
    db.setPassword('123')
    db.open()
    print(db.lastError().text())
    return True
```

**45 What is the purpose of the QSqlQueryModel class?**

Provide a read only model based on the specified SQL query

**46 Write a method belonging to the QSqlQueryModel class that will return all the rows from the product table?**

```
self.model.setQuery("Select * from products")
```

**47 What method would you invoke from the QSqlQueryModel class that will return row 2 of the specified database?**

```
self.model.record(1)
```

**48 What is the purpose of the following snippet of code:**

```
record.value("column_name") :?
```

Used to retrieve the value of the specified column of the current row of the database table

**49 Explain what is happening over here:**

```
cursor.execute("Select * from Products where prod_id=%d" %p)
row = cursor.fetchone()
```

A query is being executed and the result set is store to the cursor. row is assigned the value of one row from the result set.

