Unit 20: Event Driven Programming Solutions

Unit code: H/601/0453
QCF Level 4: BTEC Higher National
Credit value: 15

**Aim**
To provide learners with an understanding of the principles of event driven programming as an underpinning technological concept in the fields of programming and systems development.

**Unit abstract**
Unlike traditional programming, where the flow of control is determined by the program structure and the programmers design, the control in event driven programs is largely driven by external events and is often determined by interaction with the user. Typically, the systems involved employ pre-programmed event loops (or listeners) to continually look for information to process. Event driven programming is a very flexible way of allowing programs to respond to many inputs or events. It is used for all GUI based applications and can be found in web based multimedia as well as mobile technologies.

This unit allows learners to become familiar with the underpinning concepts of event driven programming and subsequently to develop particular skills in one chosen language. A variety of languages have the capacity to develop event driven solutions and it is not important which language is chosen as long as the skills being developed and evidenced relate to the key event driven focus.

As with all programming, a focus on developing solutions to meet identified needs is made along with one that emphasises the importance of testing and reviewing.

**Learning outcomes**
On successful completion of this unit a learner will:
1. Understand the principles of event driven programming.
2. Be able to design event driven programming solutions
3. Be able to implement event driven programming solutions
4. Be able to test and document event driven programming solutions.
Unit content

1  **Understand the principles of event driven programming**

*Characteristics*: key characteristics eg event handlers, listeners, trigger functions, event loops, forms

*Features*: key features eg flexibility, suitability for Graphical User Interface (GUI) environments, simplicity of programming, ease of development, potential portability

*Programming languages*: available languages eg Visual Studio .Net environment, Action Script, Java, C++

*Development environments*: environments eg for a given GUI, Java Runtime, mobile phones, multimedia, web based

2  **Be able to design event driven programming solutions**

*Development*: selection eg identification of programming language, identification of programming libraries, selection of development environment

*Design methodology*: options eg reuse of existing system, adaptation of code, use of open source

*Design method*: tools eg GUI template, graphical interface, design guides, state and interaction diagrams, screen layouts, data storage event procedures and descriptions, data and file structures

*Specification*: contents eg input, output, processes, user need, purpose

*Creation of application*: use of development environment eg mobile, handheld, web based, desktop, dedicated device; debugging delivery environment

*Interaction*: considerations eg exchange of data with other systems, compliance, compatibility, recognition of standards employed

3  **Be able to implement event driven programming solutions**

*Triggers*: types eg key press, alarm, system event, touch screen event, mouse click, external trigger, network event, incoming data, incoming call, Global Positioning Systems (GPS) data change

*Tools and techniques*: tools eg use of tool boxes and controls, debugging tools; techniques eg selection, loops, event handlers, triggers, listeners, objects and object properties, menus

*Data*: properties eg variables, data types, declaring variables, scope of variables, constants

*Programming*: coding eg use of methods, use of ‘traditional coding’

*Control structures*: types eg subroutines, branching, interrupts, signals

*Complexity*: multiple events; user interaction

*Errors*: handling eg management of extremes, use of system imposed statements

*IDE*: typical elements eg source code editor, compiler, interpreter, build automation tools, debugger
4 Be able to test and document event driven programming solutions

*Testing*: mechanisms eg valid declarations; debugging code; comment code; naming conventions; checking functionality against requirements; documentation

*Impact testing*: types eg range testing, input testing, load testing, system compatibility

*Onscreen help*: methods eg pop-ups, help menu, hot-spots

*Documentation*: technical documentation to include eg designs, delivery system, platform, environment, file structures, coding, constraints, maintenance requirements
## Learning outcomes and assessment criteria

<table>
<thead>
<tr>
<th>Learning outcomes</th>
<th>Assessment criteria for pass</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>On successful completion of this unit a learner will:</strong></td>
<td><strong>The learner can:</strong></td>
</tr>
<tr>
<td><strong>LO1</strong> Understand the principles of event driven programming</td>
<td>1.1 discuss the principles, characteristics and features of event driven programming</td>
</tr>
</tbody>
</table>
| **LO2** Be able to design event driven programming solutions | 2.1 design an event driven programming solution to a given problem  
2.2 identify the screen components and data and file structures required to implement a given design |
| **LO3** Be able to implement event driven programming solutions | 3.1 implement an event driven solution based on a prepared design  
3.2 implement event handling using control structures to meet the design algorithms  
3.3 identify and implement opportunities for error handling and reporting  
3.4 make effective use of an Integrated Development Environment (IDE) including code and screen templates |
| **LO4** Be able to test and document event driven programming solutions | 4.1 critically review and test an event driven programming solution  
4.2 analyse actual test results against expected results to identify discrepancies  
4.3 evaluate independent feedback on a developed event driven programme solution and make recommendations for improvements  
4.4 create onscreen help to assist the users of a computer program  
4.5 create documentation for the support and maintenance of a computer program. |
Guidance

Links to National Occupational Standards, other BTEC units, other BTEC qualifications and other relevant units and qualifications

The learning outcomes associated with this unit are closely linked with:

<table>
<thead>
<tr>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit 6: Software Design and Development</td>
<td>Unit 18: Procedural Programming</td>
<td>Unit 35: Web Applications Development</td>
</tr>
<tr>
<td>Unit 14: Event Driven Programming</td>
<td>Unit 19: Object Oriented Programming</td>
<td>Unit 39: Computer Games Design Development</td>
</tr>
<tr>
<td>Unit 15: Object Oriented Programming</td>
<td>Unit 21: Software Applications Testing</td>
<td>Unit 40: Distributed Software Applications</td>
</tr>
<tr>
<td>Unit 16: Procedural Programming</td>
<td>Unit 22: Office Solutions Development</td>
<td>Unit 41: Programming in Java</td>
</tr>
<tr>
<td></td>
<td>Unit 23: Mathematics for Software Development</td>
<td>Unit 42: Programming in .NET</td>
</tr>
</tbody>
</table>

This unit has links to the Level 4 and Level 5 National Occupational Standards for IT and Telecoms Professionals, particularly the areas of competence of:

- Software Development.

Essential requirements

Whilst some event driven languages are commercially available, there are also free languages available incorporating an advanced set of event driven features deployed on many platforms. Centres must ensure that in the case of mobile platforms the applicable free emulators are available or where security policies dictate, local workstations are equipped with virtualised operating systems containing the programming environment.

Learners must have access to facilities, which allow them the opportunity to fully evidence all of the criteria of the unit. If this cannot be guaranteed then centres should not attempt to deliver this unit.

The learner must develop an application that is event driven and may work on a range of platforms, therefore it may be web based, GUI based, games console or a deliverable for a mobile platform amongst many other solutions.
Resources

Books

Websites
http://java.sun.com/docs/books/tutorial/
http://tech.miradigm.com/proc_quick.php
http://visualbasic.freetutes.com/
www.developerfusion.com/tag/vb.net/

Employer engagement and vocational contexts

Working with a local programming-based organisation or using internet-based open source projects would enhance the learners’ experience and offer a relevant vocational context.