

<b>Module Title</b> <b>Interface Design</b>  <b>Keywords</b> User Interface, Usability, Event-driven Software.	Reference CM2006 SCQF Level SCQF 8 SCQF Points 15 ECTS Points 7.5 Created May 2002 Approved April 2005 Amended August 2007 Version No. 4
--	---

## This Version is No Longer Current

The latest version of this module is available [here](#)

### Prerequisites for Module

None, in addition to SCQF8 entry requirements.

### Corequisite Modules

None.

### Precluded Modules

None.

### Aims of Module

To provide the student with the knowledge and skills required to develop graphical user interfaces using an event-driven programming language.

### Learning Outcomes for Module

On completion of this module,

### Mode of Delivery

Principles and techniques used are introduced during lectures through notes, slides and examples.

Tutorials are used to emphasise the design aspects of the course material. Practical laboratory work reinforces and extends the material delivered in lectures and is student centred, building on designs developed during tutorials.

### Assessment Plan

	Learning Outcomes Assessed
Component 1	1,2,3,4,5

Component 1 - Coursework

### Indicative Bibliography

students are expected to be able to:

1. Apply user-centred methods to establish user's requirements from a project brief.
2. Apply the principles of user-centred design to the design of a graphical user interface.
3. Develop prototypes of the design using an event-driven programming language.
4. Develop the code performing underlying functionality.
5. Evaluate an application to ensure it meets the users' requirements.

1. LEWIS, C. and RIEMAN, J., 2008. Task-Centered User Interface Design: A Practical Introduction. Shareware book: available from: <ftp.cs.colorado.edu>.
2. LIDWELL, W., HOLDEN, K. and BUTLER, J. 2010. Universal Principles of Design. Rockport
3. PREECE, J., ROGERS, Y. and SHARP, H. 2007. Interaction Design: Beyond Human-Computer Interaction. Wiley

## **Indicative Module Content**

Event-driven software. Graphical User Interface components. User classes. Usability. User Requirements Analysis, Task Analysis, Object-Action Model, Dialogue Design. Visual Design. Prototyping. End-user evaluation.

## **Indicative Student Workload**

<i>Contact Hours</i>	Full Time
Lectures	18
Tutorials	6
Laboratories	24
Assessment	15

### *Directed Study*

Directed Reading	47
------------------	----

*Private Study*  
Private Study