

## **INF1002 Object-Oriented Software Development - Module Guide**

### ***Autumn 2007***

Module Leader: Brian Farrimond (farrimb@hope.ac.uk, FML 206, 0151 291 3606)

### **Introduction**

Welcome to the 30 credit Object Oriented Software Development module.

Modern software such as used in information systems is built using the systematic approach of modelling the requirements then implementing the model in an appropriate computing language. This module shows how information systems can be analysed and their requirements modelled with the Unified Modelling Language (UML). The resulting object oriented design is then implemented in a suitable programming language. The object-oriented paradigm will be at the heart of this module but other methodologies will be outlined for comparison purposes.

### **Aims**

This module aims to

- give you an appreciation of the issues and processes of modern software development;
- equip you with the tools and knowledge to carry out successful analysis and design activities;
- enable you to implement successfully object oriented designs in an appropriate programming language

### **Learning outcomes**

On successful completion of this module, you will be able to:

- LO1 Analyse information systems and model their requirements using Unified Modelling Language.
- LO2 Critically evaluate a model of a system.
- LO3 Compare and contrast aspects of the Object Oriented paradigm with other methodologies.
- LO4 Implement an Object Oriented Design in a suitable programming language.
- LO5 Select and use appropriate components from object/class libraries.

**The module web site contains all the latest  
information:**

**[http://hopelive.hope.ac.uk/imc/level\\_i/oosd](http://hopelive.hope.ac.uk/imc/level_i/oosd)**

## Syllabus

### ***Systems and Modelling***

Types of modelling/types of systems. Context: historical background, problems in software development, development of methods and tools for information modelling. The role of modelling in overall view of design. Abstraction.

### ***Object-Oriented Modelling***

Dynamic and static models. Using OO models in analysis and design: Identification of use cases and scenarios, classes, objects, associations, links, inheritance, operations, attributes.

### ***Object Orientation in context***

Outline of other methodologies: function driven, data driven. To what extent does Object Orientation meet its claims?

### ***Object Oriented programming***

Constructs. Encapsulation, Classes, Objects, Inheritance, Polymorphism, Communication between objects (messages). Implementation of an Object Oriented Design in a suitable language, such as Java.

## Module Resources

### ***Course books***

There are two books that you will need to buy for this module:

#### **Beaumont and Farrimond, Object Oriented Systems Analysis and Design, Liverpool Hope 2004**

This book has been written by the module team specifically for this module. You will be told how you can buy copies by your tutors. Cost per copy will be £4.

#### **Barnes and Kolling, Objects First with Java, 3rd Edition, Pearson, Prentice Hall.**

This book is available at all good bookshops and on-line via Amazon etc. Copies are also available in the Hope library.

### ***Course software***

The systems analysis part of the module is heavily dependent on diagrams. You will need to submit diagrams electronically for your assessment. You may produce these diagrams within Microsoft Word or use a drawing package such as **Microsoft Visio**, available on the college licence which has tools for drawing UML diagrams rather more easily.

The Java book has an associated free Java development environment called **BlueJ** available on the accompanying CD or on-line at <http://www.bluej.org/>.

To install BlueJ you **should first** install the Java Software Development Kit (JSDK) which is contained on the CD, in the j2sdk\windows folder.

BlueJ is primarily intended as a learning environment. Towards the end of the course, you will be using a professional development environment called Net Beans. This is free and can be downloaded from: <http://www.netbeans.org/downloads/> .

### **Time to be spent on this module**

Lecture / Seminars	25 hours
Practical laboratories	50 hours
Preparatory reading / Independent study	81 hours
Coursework	120 hours
Revision	24 hours
<b>Total</b>	<b>300 hours</b>

Each week you will be expected to:

- attend 1 lecture of 1 hour, a practical lab of 2 hours;
- work independently for approximately 9 hours.

## Assessment

Assessment	Description	Learning outcomes	Weighting
Portfolio of exercises and quizzes	You will be given sets of exercises to work on during the module. Samples of these will be placed in your portfolio as evidence of engaging with your learning. The portfolio will also contain a three page Java Survival Guide that you will develop as you learn. Quizzes will take place over the duration of the course as indicated in the schedule given in this module guide.	LO1, LO4	10 %
Coursework 1 (CW 1)	You will develop an information system using Java. It will consist of an underlying engine that you will submit soon after Christmas and a GUI written in NetBeans that you will submit towards the end of the module.	LO4, LO5	20%
Coursework 2 (CW 2)	You will modify and/or extend a given model.	LO1, LO2	20%
Coursework 3 (CW 3)	You will carry out an initial analysis of a real system and document it with UML.	LO1	20%
Time Constrained assignment 1 (TCA 1)	Java Test Duration 90 mins	LO4, LO5	15%
Time Constrained assignment 2 (TCA 2)	Systems Analysis Test Duration 90 mins	LO1, LO2, LO3	15%

**Schedule (Provisional - see web site for latest version)**

<b>Weeks</b>	<b>Main topic</b>	<b>Lecture</b>	<b>Practical labs</b>	<b>Private Study</b>	<b>Submitted Work</b>
1 Mon 1 Oct	Software Development Concepts: OOSAD Ch 1, Ch 2	Brief Module Intro OOSAD - Chapter 2: The story so far  Issue of Java Portfolio Exercises	Systems and Models Exercises based on OOSAD Chapter 1  <b>Complete Quiz (formative) Exercises in OOSAD Chapters 1 and 2</b>	Read OOSAD – Chapters 1 & 2: for consolidation. Make notes of any questions.  <b>Complete exercises in OOSAD Chapters 1 and 2</b>  You should obtain a copy of the Java textbook, install Java and work through the first chapter in the Java book by the beginning of week 2.	
2 8 Oct	Java Ch 1	Java Ch 1, objects and classes	<b>Quiz 1 on OOSAD Chapter 2</b> Using BlueJ. Workshop on Chapter 1 exercises	Read Java Ch 2  Work on Java Chapter 1 exercises for Portfolio	
3 15 Oct	Java Ch 2	Java Ch 2, first part: class definitions	<b>Quiz 2 on Java Chapter 1</b> Workshop on Chapter 2 exercises	Work on Java Chapter 2 exercises	
4 22 Oct	Java Ch 2	Java Ch 2, second part: class definitions	Workshop on Chapter 2 exercises	Work on Java Chapter 2 exercises	
5 29 Oct	Java Ch 3	Java Ch 3, first part: object interaction Issue of CW 1. Introduction to Part A	<b>Quiz 3 on Java Chapter 2</b> Workshop on Chapter 3 exercises	Work on Java Chapter 3 exercises	
6 5 Nov	Java Ch 3	Java Ch 3, second part: object interaction	Workshop on Chapter 3 exercises	Work on Java Chapter 3 exercises  Work on Portfolio Exercises	

7 12 Nov	Java Ch 4	Java Ch 4, grouping objects	<b>Quiz 4 on Java Chapter 3</b> Workshop on Chapter 4 exercises	Work on Java Chapter 4 exercises Work on Portfolio Exercises	
8 19 Nov	Java Ch 4	Java Ch 4, grouping objects Introduction to CW 1 Part B	Workshop on Chapter 4 exercises	Work on Java Chapter 4 exercises Work on Portfolio Exercises	
9 26 Nov	Java Ch 5	Java Ch 5: Using class libraries	<b>Quiz 5 on Java Chapter 4</b> Workshop on Chapter 5 exercises	Work on Java Chapter 5 exercises Work on Portfolio Exercises	
10 3 Dec	Java Ch 8	Java Ch 8: Inheritance	Workshop on Chapter 8 exercises	Work on Java Chapter 8 exercises Work on Portfolio Exercises	
11 10 Dec	Java Ch 8	Java Ch 8: Inheritance Introduction to CW 1 Part C	Workshop on Chapter 8 exercises. Revision issues	Work on Java Chapter 8 exercises Work on Portfolio Exercises	
		<b>Christmas Vacation</b>			
12 7 Jan	Java Ch 9 Revision	Java Ch 9: Inheritance	Workshop on Chapter 9 exercises. Revision issues		
13 14 Jan	Java test	<b>Java Test TCA1 (Wednesday class)</b>	<b>Demonstrate Portfolio exercise and submit on CD with Java Survival Guide</b>	Read OOSAD Chapter 3	<b>Submit CW 1.1</b>
14 21 Jan	OOSAD Ch 3	OOSAD Ch 3, introductory case study	Exercises based on OOSAD Chapter 3	Exercises on OOSAD Chapter 3. Read OOSAD Chapter 4: Class models	
15 28 Jan	Class Models OOSAD Ch 4	Lecture on class models	<b>Quiz 6 on OOSAD Ch 3</b> Group exercises creating and amending class models	Exercises based on OOSAD Chapter 4	
16 4 Feb	Class Models OOSAD Ch 4	Lecture on issues in class models	Exercises based on OOSAD Chapter 4: class models	Exercises based on OOSAD Chapter 4 Read OOSAD Chapter 5: Use cases	

17 11 Feb	Use cases OOSAD Ch 5	Lecture on use cases	<b>Quiz 7 on OOSAD Ch 4</b> Exercises based on OOSAD Chapter 5	Exercises based on OOSAD Chapter 5. Read CW 2	
18 18 Feb	CW 2	Issue of CW 2 Lecture on CW 2 and intro to CW 3.	<b>Quiz 8 on OOSAD Ch 5</b> Exercises based on OOSAD Chapter 5.	Work on CW 2	
19 25 Feb	CW 2	Issues on CW 2	Issues on CW 2	Work on CW 2	
20 3 Mar	Java NetBeans	Lecture on NetBeans Issue and intro to CW 1.4.	Using NetBeans	Exercises on NetBeans. Work on CW 3.	<b>Submit CW 2</b>
21 10 Mar	Java NetBeans	Issue of CW 3 Feedback on CW 2	Using NetBeans for CW 1.4	Work on CW 3, CW 1.4	
		<b>Easter Vacation</b>			
22 7 Apr	CW 3	Issues on CW 3, CW 1.4	Issues on CW 3, CW 1.4	Work on CW 3, CW 1.4	
23 14 Apr	Issues	Issues on CW 3, CW 1.4	Issues on CW 3, CW 1.4	Work on CW 3, CW 1.4	
24 21 Apr	Revision	OOSAD Test TCA2 revision	OOSAD Test TCA2 revision	Prepare for Test TCA2	<b>Submit CW 3</b> <b>Submit CW 1.4</b>
25 28 Apr	Revision and test	<b>OOSAD Test TCA2</b>			

**Submitted work must be handed in to the Deanery office by 3 pm on Friday of the designated week.**