Liverpool John Moores University

Title:	SOFTWARE DEVELOPMENT JAVA
Status:	Definitive
Code:	**7012ONLINE** (103119)
Version Start Date:	01-08-2017
Owning School/Faculty:	Computer Science
Teaching School/Faculty:	Computer Science

Team	Leader
Denis Reilly	Y

Academic Level:	FHEQ7	**Credit Value:**	15	**Total Delivered Hours:**	30
Total Learning Hours:	150	**Private Study:**	120		

Delivery Options
Course typically offered: Runs Twice - S1 & S2

Component	**Contact Hours**
Lecture	12
Seminar	12
Tutorial	6

Grading Basis: 40 %

Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Technology	AS1	Object-oriented Analysis, Design and Implementation	50	
Technology	AS2	Class responsibility collaboration (CRC) based Software Development	50	

Aims

The course will developthe necessary skills for the development of object-orientated applications using the Java programming language. Students will work cooperatively in groups and demonstrate the skills required to engineer Java-based software applications from initial specification, through to implementation, testing and *documentation.*

Learning Outcomes

After completing the module the student should be able to:

1 Critically analyse the use of object-oriented principles in the design of software applications.

2 Use object-oriented principles to produce UML specifications of software applications

3 Demonstrate mastery of UML specifications using the Java programming language.

4 Edit and document Java-based applications

5 Demonstrate the ability to work in small teams to distribute and manage the tasks required of points 2, 3 and 4.

Learning Outcomes of Assessments

The assessment item list is assessed via the learning outcomes listed:

Object-oriented Analysis	1	2	3		
Software Development	1	2	3	4	5

Outline Syllabus

Foundations of object-orientation.
Anatomy of Java classes – fields, constructors, methods.
Objects and classes – what is an object, object state, objects as parameters.
Object interaction – method invocation, objects calling objects.
Designing classes – responsibility-driven design.
Application structures – inheritance, subtyping, polymorphism.
Abstraction techniques – simulation, abstract classes, interfaces.
Handling errors – defensive programming, exceptions.
Designing applications – analysis and design, CRC cards, scenarios, class design, documentation, group cooperation.
Case Study – design of a chosen application.
Java-based user-interfaces – AWT and Swing APIs

Learning Activities

Online Lectures will be accompanied by practical online sessions. Students will be required to work in small groups to complete tasks, thereby encouraging communication and projects management skills

Notes

The module lectures, tutorials and labs will use the BLUEJ development tool, which is a GUI-based development aid based on UML. All online activities are scheduled.