Liverpool John Moores University

Title:	SOFTWARE DEVELOPMENT WITH JAVA
Status:	Definitive
Code:	**7067COMP** (120325)
Version Start Date:	01-08-2018

Owning School/Faculty:	Computer Science
Teaching School/Faculty:	Computer Science

Team	Leader
Denis Reilly	Y

Academic Level:	FHEQ7	**Credit Value:**	20	**Total Delivered Hours:**	36
Total Learning Hours:	200	**Private Study:**	164		

Delivery Options

Course typically offered: Semester 1

Component	Contact Hours
Lecture	12
Practical	24

Grading Basis: 40 %

Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Technology	AS1	Coursework – Object-oriented Design and Implementation.	50	
Technology	AS2	Group Coursework – Team-based Software Development.	50	

Aims

The course will develop the necessary skills for the development of object-oriented 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   Explain the use of object-oriented principles in the design of software applications..
2   Use object-oriented principles to specify and design software applications.
3   Work in teams to implement object-oriented designs using the Java programming language.
4   Work in teams to test and document Java-based applications.
5   Evaluate the use of object-oriented development techniques.

Learning Outcomes of Assessments

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

OO Design and Implementation	1	2	
Software development	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
Advanced topics – e.g. generics

Learning Activities

Lectures will be accompanied by practical lab-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 loosely based on UML. The group coursework will be completed in groups of two students and peer assessment will be used to assess individual student effort.