Liverpool John Moores University

Title: SOFTWARE DEVELOPMENT FOR COMPUTER FORENSICS
Status: Definitive
Code: **7078COMP**   (120610)
Version Start Date: 01-08-2019

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

Team	Leader
Rubem Pereira	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
Tutorial	24

Grading Basis: 40 %

Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Technology	AS1	Coursework - Design, Implement, and Test Software to Solve a Digital Forensic Challenge.	100	

Aims

The course will develop the skills of students to create scripts that address digital forensic needs. Students will consider real-world digital forensic problems and demonstrate the skills required to engineer applications that solve these problems from initial specification through to implementation and testing.

Learning Outcomes

After completing the module the student should be able to:

1 Explain the use of interpreted scripting languages in digital forensics.
2 Use functional and object-oriented principles to specify and design software applications for digital forensics.
3 Work individually to implement software designs that address digital forensic challenges using interpreted scripting languages.
4 Work individually to test and document interpreted scripting language applications.
5 Evaluate the use of functional and object-oriented development techniques using interpreted scripting languages in digital forensics.

Learning Outcomes of Assessments

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

Software Development   1   2   3   4   5

Outline Syllabus

Foundations of interpreted scripting languages.
Role of scripting within digital forensics.
Functional programming techniques - first-class functions, closures, lambda functions.
Containers and iteration - lists, maps, iterators, generators.
Classes and objects - what is a class, what is an object, object state.
Object interaction - method invocation, special methods, objects as parameters.
Interacting with the outside world - input and output, streams, network programming.
Accessing common file formats - structured markup, images, documents.
Database programming - database clients, querying data.
Reporting and visualisation - creating reports, generating visualisations.
Designing applications - analysis and design, agile software development, unit testing.
Case study - design and implementation of a chosen digital forensic application.
Advanced topics - concurrency and parallelism, cross-platform development, etc.

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 be conducted with interpreted scripting languages, development environments, and libraries that are widely used within the digital forensics community.