|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: 3D Interactive Media Development | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries, School of Computer Science and Engineering | | |
| Module Leader(s): Pumudu Fernando | | |
| Extension: | Email: pumudu.f@iit.ac.lk | |
| Host course and course leader: BSc Computer Science | | |
| Status: Core - BSs Digital Media Development; Option - BSc Computer Science, BSc Computer Games; Westminster Plus Elective | | |
| Subject Board: COMENG | | |
| Pre-requisites: none | Co-requisites: none | |
| Study abroad: No alternative CW required | | |
| Special features: IIT Franchise Module | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an  arrangement with an organisation(s) other than the University of Westminster: No | | |
| Summary of module content  This module introduces students to concepts of 3D user interfaces for interactive media. It covers essential topics including 3D modelling principles, methods and techniques, complex object creation, deformation and transformations, texture mapping, colour and lighting. It uses an industry standard games engine to demonstrate 3D animation concepts, properties, controllers and scripting to create interactive media content. This is part of the “Usability and Interaction” and “Games and Computer Graphics Development“ themes for Computer Science, but is open to all courses with no pre-requisite. Supported coursework path is the production of an interactive rich media product. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Demonstrate an applied intermediate understanding of 3D modelling concepts and the underlying geometrical principles and apply appropriate techniques and tools to create 3D interactive media;

LO2 Incorporate 3D content in an industry standard games platform/engine;

LO3 Demonstrate an applied intermediate understanding of 3D animation concept, properties, controllers and use of scripting techniques to create 3D animations;

LO4 Demonstrate an applied understanding of scripting techniques to incorporate interactive and navigation behaviours for 3DUI concepts;

LO5 Work individually, as well as an effective team member to develop a common 3D

interactive media product;

LO6 Communicate design concepts by oral and visual means and provide documentation for a 3D interactive media product in written form.

**Course outcomes the module contributes to**

BSc Digital Media Development L5.1, L5.3, L5.6, L5.7, L5.11

**Indicative syllabus content**

* + - 3D User Interfaces (UIs) and interaction styles, introduce applications of 3D UIs, gamified interfaces, mobile and web applications, etc. Discuss principles of designing Usable 3D UIs. Understand the design space.
    - 3D world representation and 3D space transformation. Expansion of Game Design and Asset Creation.
    - Key concepts of 3D modelling, primitives: Expansion of Game Design and Asset Creation. Polygonal mesh representation, intermediate creating and editing a scene, polygon editing and hard edge modelling, parametric curves, NURBs and spline- based modelling concepts, lathing, extruding and Boolean modelling techniques to create more complex models, groups and hierarchies, construction history.
    - Understanding Visual effects, shading (composing and colour grading), texturing mapping (UV texture layout, coordinates, mapping) and paint effects. Brief introduction to lighting (illumination model, light types and properties) and shading, Expansion of Game Design and Asset Creation.
    - Preparation for animation, basic skeletons (not full body character) and basic rigging

techniques (not to the extent of character animation).

* + - Animation techniques, timeline based animation; motion paths; basic relationships:

constraints, set driven keys. Standard 3D UI for 3D interactive games engines.

* + - Interactivity, control 3D geometry and animation properties with the use

of appropriate scripting language.

* + - Navigation, control a 3D camera system and enable various POV systems and interactions. Combine navigation with animation as required.
    - State Changes
    - Current 3D UI Research, investigation of latest concepts and work, especially

from <http://ieeevr.org/>

* + - Current 3DUI platforms and delivery, build to platforms and begin to take XR systems and hardware/software into account, likely WebXR and especially AR, leading to the XR Multimodal Interaction module in semester 2.

**Teaching and learning methods**

This module is taught using a combination of lectures or workshops and laboratory-based tutorial sessions or workshops. The lectures or workshops include demonstration of the syllabus and the students are given a framework based on which they can classify and evaluate the course materials in their subject. During tutorials or workshops, students are introduced to industry-standard tools that are used for the development of a 3D scene and 3D animated and interactive content, offering experience of applying knowledge acquired during lectures through practical exercises and case studies. Students are provided with a set of practical exercises to be completed both during and outside the scheduled tutorial times. Immediate feedback on proposed solutions to the exercises is provided in the tutorials with discussion of problems providing formative feedback on students’ progress and understanding.

Students will have the opportunity to present their design concept and their proposed approach based on which they will be provided with formative feedback and update their design. Students will have the opportunity to work in small groups as well as individually, to consolidate the concepts covered and enhance their practical skills in research, writing, speaking and presenting. A contract with task allocation between team members and milestones for the group project will be used as a mechanism to manage groups and tasks and milestones completion will be a parameter to be reflected in marking to ensure fair assessment. Strong connections are made with the students’ learning skills requirements elsewhere in the course.

There are 152 hours allocated to Independent study, this forms a very important part of the module where as a student you are expected to use this time to prepare and develop your skills and maximise your time by using the practical classes as a point of contact to clarify any issues you may have, thus maximising your learning. Typically, as a guide you should break down your time for independent study in the following way:

Webinars: Guided Independent Study – 24 hours Preparation of Assessment – 12 hours per CW Preparation of Peer Review – 4 hours per CW Development work – 60 hours per CW

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 12 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 12 |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled | 6 |
| Practical Classes and workshops | Scheduled | 6 |
| Supervised time in studio/workshop | Scheduled | 6 |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled | 6 |

|  |  |  |
| --- | --- | --- |
| **Total Scheduled** |  | **48** |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | **200** |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

The coursework rationale is to introduce the students to the practical nature of 3D interactive media development. It realizes the above by setting two pieces of coursework that assess student knowledge and understanding in different aspects of composing a 3D interactive media product:

* + Coursework 1 - assesses the student skills in working as a team to plan and visualize a given 3D interactive media concept that addresses specific technical requirements, translate these into a practical sequence and compose the required 3D scene and content (the content will be later animated and interactive by adding appropriate behaviours to it) using industry standard tools. In addition, it assesses the student’s skills presenting and communicating their design concept efficiently. (Learning outcomes 1, 2, 5, 6)
  + Coursework 2 - assesses the student practical ability to use appropriate scripting language to create required animation and interactivity to a proposed 3D interactive media product. In addition, it assesses the student’s skills documenting their implementation in a report using a professional style and their entrepreneurial skills by the creation of a working video demo of their project. (Learning outcomes 3-6).

Both pieces of coursework are assessed by means of an extended tutorial exercise, to be completed both during and outside tutorials. Milestones are set and coursework components are submitted progressively and discussed during tutorials. There are numerous of benefits following this process: it allows formative feedback to be provided; it allows reflection and produces progressive iterations of the design; it aids continuous engagement; it makes possible to track engagement. Summative assessment is applied on the final iteration of the 3D interactive media product.

**Assessment criteria**

The student will demonstrate achievement of the learning outcomes by being able to:

* + Compose a 3D interactive media product following given technical requirements without restricting creativity;
  + Using appropriate tools and scripting language to create 3D animated, interactive media content, smoothly integrate it in a 3D interactive media product and deploy it for multiple platforms and devices;
  + Demonstrate professional behavior in identifying roles, delegating tasks to group members, recognizing obligations, working individually, as well as a member of a team, in the development of a quality assured solutions to a medium-scale interactive media product;
  + Communicate design concepts by oral and visual means;
  + Write documentation of a 3D interactive media product.

To achieve higher marks than the threshold, students must demonstrate a greater degree of expertise and knowledge, and in particular to demonstrate knowledge of economic programming skills, creative technology skills, material beyond that covered in class, as well as the ability to effectively work in a group.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| ***Group coursework 1*** | *50* | *30* |  | *Formative,Coursework Presentation* |
| ***Group coursework 2*** | *50* | *30* |  | *Coursework - Report* |

**Synoptic assessment**

This module draws on technical, analytical, research and presentation skills learned elsewhere in the course. This module therefore engenders synoptic learning and its assessment is inherently synoptic.

**Sources**

**Link to the online reading list**

[https://rl.talis.com/3/westminster/lists/FFA649D0-172E-2F19-C52B-](https://rl.talis.com/3/westminster/lists/FFA649D0-172E-2F19-C52B-EFD7D3DB5C18.html?lang=en-GB&amp;login=1)

[EFD7D3DB5C18.html?lang=en-GB&login=1](https://rl.talis.com/3/westminster/lists/FFA649D0-172E-2F19-C52B-EFD7D3DB5C18.html?lang=en-GB&amp;login=1)

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Advanced Client-side Web Development | | |
| Module code: | Credit level: 4 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Theja Perera | | |
| Extension: | Email: theja.p@iit.ac.lk | |
| Host course and course leader: BSc Computer Science | | |
| Status: Option - BSc Computer Science, BSc Business Information Systems | | |
| Subject Board: COMENG | | |
| Pre-requisites:  Web Design and Development | Co-requisites: | |
| Study abroad: | | |
| Special features: | | |
| Access restrictions: | | |
| Are the module learning outcomes delivered, assessed or supported through an  arrangement with an organisation(s) other than the University of Westminster. No | | |
| Summary of module content  This module provides practical knowledge and understanding of client-side or/else front- end development programming using advanced HTML5, CSS3 and JavaScript. Client-side technologies, including HTML5 Audio and Video are covered together with a client-side scripting language, a UI and CSS framework and a client-side scripting framework. The module also covers issues pertaining to front-end security. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Be proficient in programming using markup (HTML5) and stylesheet (CSS3) languages;

LO2 Effectively incorporate frameworks such as jQuery for DOM selection, decoration and

enhancing;

LO3 Develop a responsive website using media queries;

LO4 Meet client needs by demonstrating practical skills in following specific requirements to build a highly interactive, web site with appropriate use and handling of multimedia content (audio, video);

LO5 Understand issues pertaining to front-end security.

**Course outcomes the module contributes to**

**Indicative syllabus content**

* + - **HTML5** - HTML5 specification, including audio, video, responsive HTML5 forms,

Accessibility and Web fonts.

* + - **CSS3** – Advanced Selectors, Specificity, Page layout, Advanced CSS3 techniques including special effects (3D, ribbon, transition, animation, rotation), advanced navigation effects.
    - **Building Rich User Interfaces** - Building rich user interfaces with jQuery widgets: accordion, auto-complete, menus, tabs, tooltips, spinners, progressbars and date- pickers.
    - **Media queries** - Introduction to Responsive Web Design (RWD) and media queries.
    - **JavaScript & JavaScript Framework** - Creation of interactive websites using JavaScript, the jQuery core library and jQuery UI; DOM traversal, selection and manipulation, content creation, content appending, event handling, event override.
    - **Effects** - Introduction to JavaScript effects such as clip, drop, fold, fade, highlight, puff, pulsate, slide, toggle and translate for web page enhancement.
    - **Interaction** - Interaction through making elements draggable, droppable, resizable,

selectable and sortable.

* + - **Animation** - Animation and transitions in CSS3 and/or JavaScript. Compare and

contrast both options.

* + - **Browser Developer Tools** - Explore browser development tools for performance optimisation and debugging.
    - **Security** - Explore front-end security and HTML5 vulnerabilities, cross-site scripting,

iFrames and Cookies.

**Teaching and learning methods**

The module involves a significant element of guided learning to allow the students to master technical skills and professional qualities required to develop rich, responsive front end user experiences on the Web. The lectures (2 hours weekly) include demonstration of the syllabus. Concepts, features and techniques are demonstrated by developing live example code using utilities running on a browser.

Tutorials (2 hours weekly) include a set of practical programming exercises that explore aspects of Client-side programming to be completed both during and outside the scheduled tutorial times. Immediate feedback on proposed solutions to the exercises is provided in the tutorials with discussion of problems. Tutorials are central to the learning, and will offer the opportunity to interact with fellow students and tutors to gain understanding of the topics and confidence in communicating and applying them.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 24 |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled |  |
| Supervised time in studio/workshop | Scheduled |  |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled |  |
| **Total Scheduled** |  | 48 |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | 200 |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

The summative assessment strategy involves the development of two separate applications: a web-based portfolio, using HTML5, and advanced CSS3 techniques, and a more advanced and richer client-side web application using HTML5, CSS3, JavaScript, and client-side web frameworks and plugins. The first application will be assessed by in-class test to evaluate the student understanding, while the second one will be assed by viva, which will evaluate the students’ ability to apply the technologies and techniques taught on the module to the development of advanced client-side web applications.

The first assessment (Portfolio with in-class test) is developed over the first few weeks and students are required to develop a personal web-based portfolio using specific HTML5 and CSS3 techniques by form of a series of exercises. These will allow the students to develop their knowledge of Client-side programming that shall form a portfolio. The student understanding is assessed in an accompanying in-class test (LO1, LO4).

The second assessment (coursework) is a highly interactive and responsive website that incorporates the features and techniques that are demonstrated by the syllabus coupled with a viva, through which the students will demonstrate a thorough knowledge of issues related to the solution (LO1 - LO5).

Formative assessments will be embedded within the teaching and learning activities of the module. Students will be required to regularly demonstrate their knowledge of advanced client- side web development in terms of markup, scripting and styling, and they will receive constructive feedback regularly so they can assess their progress and identify areas that require further attention.

**Assessment criteria**

The module will measure the understanding of client-side programming fundamentals:

* + - Portfolio with in-class test: A set of exercises will be submitted on which an in-class test with be based. To pass the assignment, students are expected to demonstrate competence in practical skills they have been taught. To achieve higher marks,

students must demonstrate a greater understanding of specific challenges that require the students to work as independent learners and explore further the taught topics.

* + - Coursework: A highly interactive/responsive web site with browser events and CSS and/or framework effects. The students should demonstrate proficiency in the use of scripting and CSS frameworks to build rich user interfaces and professionalism in terms of code quality and presentation as well as client-side security.

A pass level would represent the ability to demonstrate a basic understanding of the subject without necessarily providing any further insight into the field other than that gained from the lecture material.

For higher grades, the student shall be expected to concisely summarise the given problem

and disseminate the report with a high level of understanding.

For a ‘first’ level mark, the student shall be expected to also complement the taught material with that gained by independent thought and study.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| ***Portfolio with inclass test*** | *40* | *30* |  | *In-class test* |
| ***Coursework*** | *60* | *30* |  | *Coursework (web site development)* |

**Synoptic assessment Sources**

Resources will be available via Blackboard Virtual Learning Environment. This section will be

updated every year with the most up to date reading list.

Weyl, E., Lazaris, L. and Goldstein, A. (2011), HTML5 and CSS3 for the Real World, SitePoint. Duckett, J. (2014), JavaScript & jQuery, Interactive front-end Web development, Wiley.

Frain, B. (2012), Responsive Web Design with HTML5 and CSS3, Packt Publishing.

**Further reading**

Van Toll, T.J. (2014), jQuery UI in action, Manning Publications.

Marcotte E. (2015) Responsive Design: Patterns & Principles, A Book Apart.

**Link to the online reading list**

https://rl.talis.com/3/westminster/lists/25B6ADE4-20A9-2625-FB0F-71F3ED80A5E0.html

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Applied Cryptography | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Saman Hettiarachchi | | |
| Extension: | Email: | |
| Host course and course leader: BSc Computer Network Security | | |
| Status: Core - BSc Computer Network Security; Option – BSc Computer Science | | |
| Subject Board: COMENG | | |
| Pre-requisites: | Co-requisites: None | |
| Study abroad: Not available for study abroad students. | | |
| Special features: None | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an  arrangement with an organisation(s) other than the University of Westminster. No | | |
| Summary of module content:  This module, through lectures and practical activities, introduces the theory of classical and modern cryptography and its use in computing, with an emphasis on the practical applications of cryptography to secure computer networks and systems. A systematic study of the fundamental cryptographic principles provides the foundation for an applied perspective to the subject. Practical work based on examples using popular cryptographic tools and APIs helps to develop a methodology for using cryptographic techniques to design and construct security solutions in key application domains. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Comprehend and apply fundamental concepts in cryptography;

LO2 Apply knowledge of applied cryptography to the design and evaluation of computing and networking systems;

LO3 Evaluate the main modern cryptographic approaches and their applications;

LO4 Critically assess and compare the strengths, limitations and functionality of cryptographic systems;

LO5 Design and develop an application by using appropriate cryptographic tools and methods.

**Course outcomes the module contributes to**

BSc Computer Network Security L5.2, L5.3, L5.4, L5.6

**Indicative syllabus content**

**Introduction to classical cryptography**: Cryptographic systems; mathematical foundations; classical cryptographic techniques; secret key principles; block/stream ciphers.

**Modern cryptography:** Symmetric and asymmetric encryption algorithms, hashing algorithms, key distribution, public/private key principles, public key infrastructure, digital signatures and certificates. Cryptographic advancements.

**Applications of cryptography:** System, application and network implementations. Cryptographic protocol design. Performance and efficiency of cryptographic techniques. Cryptography for common distributed computing applications: The Internet, web sites and email. Steganography.

**Attacks against cryptographic systems**: Techniques and mitigation.

**Teaching and learning methods**

Students will attend lectures, practical sessions and supervised workshops. The lectures (typically 2 hours) are the primary means of introducing the relevant concepts and theory. The lectures will be used to introduce each topic, defining different cryptographic systems and give essential practical examples to ensure the process of evaluating different cryptographic techniques and develop code implementing the cryptographic principles covered. The practical and seminars sessions (typically 1.5 hours) reinforce these concepts through practical application and scenario-based inquiry directed according to problem solving exercises in a computer laboratory.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 20 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 16 |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled | 8 |
| Supervised time in studio/workshop | Scheduled | 4 |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work based learning | Scheduled |  |
| **Total Scheduled** |  | 48 |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | 200 |

\*the hours per activity type are indicative and subject to change.

During independent study time, the student will be expected to read through lecture notes (40 hours), solve all set problems and complete assessed tasks (50 hours), and reinforce via practical work in the (open-access) laboratories the concepts covered in lectures (40 hours).

The remaining time is expected to be used on practicing for the group presentation and revising for the exam.

**Assessment rationale**

Practical abilities will be assessed by means of programming tasks involving various lab exercises.

The Practical Programming exercise will examine individual work given a programming problem implementing cryptographic techniques.

The second assessment component is a group-based presentation to assess research and presentation skills, demonstrating theoretical and technical ability within an area of cryptographic application. The nature of group assessment supports the development of essential communication, collaborative and presentation skills.

Labs will be given with exercises to formatively assess the software implementation of security features. Formative feedback will be given progressively to develop the student’s ability to advance the programming and environmental skills gained from other programming modules.

A final written exam will cover all theoretical and implementation aspects addressed in the module.

The two formative components in the module are:

Workshops and seminars will be given with set exercises to formatively assess the application of the module material. Formative feedback will be given progressively to develop the ability to advance and broaden critical analysis skills.

Practice exam questions will be set to periodically formatively assess understanding and application of the implementation and theoretical aspects of cryptographic systems.

**Assessment criteria**

To pass the module, students will have to demonstrate an understanding of cryptographic techniques and systems and to develop skills in the implementation of cryptography in an application programming language. Higher levels of achievement will be characterized by the ability to work independently to identify and resolve problems; competent group work skills; demonstrate higher levels of analytical skill in how to identify weaknesses within a system and apply a suitable cryptographic solution; and programming competence.

Coursework 1 will primarily focus on programming techniques covering LO5.

Coursework 2 will be a group presentation that covers LO2, LO3, LO4 and LO5.

The Exam will assess the students understanding and the fundamental concepts of cryptography and its implementation within systems covering LO1, LO2, LO3 and LO4.

The two formative components in the module are:

Problem-based exercises; problem sets of questions will be set on-line and formative feedback will be given to assist students in their learning covering LO2, LO4 and LO5.

Practice exam questions; these will be set to give the opportunity to assess their understanding of the theoretical aspects of the module in preparation for the final exam covering LO1, LO2, LO3 and LO4.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| *Coursework 1* | *20* | *30* |  | *Practical skills assessment* |
| *Group Presentation* | *30* | *30* |  | *Practical* |
| *Examination* | *50* | *30* |  | *Written (Closed Exam)* |

**Synoptic assessment**

This module draws upon technical skills in programming and security principles being learned elsewhere in the course and developing cryptographic knowledge. By integrating and applying these, the important links between component algorithmic design, implementation and performance characteristics for cryptographic systems is emphasised. This module therefore engenders synoptic learning and its assessment is inherently synoptic.

**Sources**

**Essential reading list**

Mel, H. and Baker, D., *Cryptography Decrypted*, Addison Wesley, 2001, ISBN: 978- 0201616477.

Stallings, W., *Cryptography and Network Security: Principles and Practices*, 6th Edition,

Prentice-Hall, 2013, ISBN: 978-0133354690.

Stamp, M., *Information Security: Principles and Practice, 2nd Edition,* Wiley, 2011, ISBN: 978-0470626399.

Trappe, W. and Washington, L., *Introduction to Cryptography with Coding Theory*, 2nd Edition, Pearson, 2006, ISBN: 978-0131862395.

**Link to the online reading list**

<https://rl.talis.com/3/westminster/lists/D0D9CB15-1246-07AB-8510-3DEECB741F46.html>

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Business Analytics | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Alroy Mascaranghe | | |
| Extension: | Email: | |
| Host course and course leader: BSc Business Information Systems | | |
| Status: Core - BSc Data Science and Analytics; Option - BSc Business Information Systems, BSc Computer Science, BEng Software Engineering, BSc Computer Network Security | | |
| Subject Board: | | |
| Pre-requisites: None | Co-requisites: None | |
| Study abroad: | | |
| Special features: None | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an  arrangement with an organisation(s) other than the University of Westminster. No | | |
| Summary of module content  This module introduces students to the Operational Research (OR) techniques, commonly used for business analytics, such as Linear programming, forecasting, simulation and decision making. It helps students to develop and analyse analytical models that support making effective business decisions. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Understand a business problem and formulate it analytically and in symbolic form, so as

to facilitate their analysis and solution;

LO2 Demonstrate knowledge of main analytical methods, and ability to evaluate critically their appropriateness to solving problems;

LO3 Demonstrate competency in applying analytical methods to solve business problems,

and appraise the effect of assumptions on analytical modelling and output analysis;

LO4 Explore data to identify trends, patterns and relationships through the application of

appropriate techniques and tools;

LO5 Effectively communicate analytical models and analysis with accuracy and clarity to support decision making.

**Course outcomes the module contributes to**

BSc Data Science and Analytics L5.2, L5.3, L5.4, L5.5, L5.7

**Indicative syllabus content Linear Programming (LP)**

* + Introduction to model building in OR.
  + Applications of LP graphical solution.
  + Using Excel’s solver to obtain solution to LP problem.
  + Sensitivity analysis.

**Forecasting**

* + Time-series analysis: Smoothing methods, Methods for trend and seasonality, Accuracy measures.
  + Causal method: Simple linear regression, coefficient of determination.

**Simulation**

* + Monte-Carlo simulation.
  + Discrete event simulation.
  + Simulation experiments and output analysis.

**Decision making methods**

* + Decision problem and decision making.
  + Scoring models.
  + Payoff tables and Decision trees.
  + Probabilistic and non-Probabilistic methods.

**Teaching and learning methods**

The module will be taught by a mixture of lectures, supervised computer laboratory sessions and self-directed study exercises. Approximately 50 % of the staff contact time will be spent on lectures and the remaining time will be spent on supervised computer laboratory sessions. The lectures will be used to introduce the various concepts and techniques of the module’s topics. During the supervised computer laboratory / workshop sessions students will have the chance to work individually or in smaller groups and to use software tools and packages to solve real life problems. It is envisaged that most of the topics covered will be underpinned with directed self-study material.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 24 |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled |  |
| Supervised time in studio/workshop | Scheduled |  |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled |  |
| **Total Scheduled** |  | **48** |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | **200** |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

The module will be assessed by an individual coursework and an exam.

The individual coursework will assess students’ overall knowledge and facilitate assessing them on their ability to model and solve problems with appropriate forecasting and/or simulation techniques, develop spreadsheet tools, and analyse and discuss their solutions. Through this assessment, the achievement of learning outcomes 2 – 5 will be assessed.

In terms of learning outcomes, the exam will assess the ability of students to formulate a linear programming (LP) problem, solve an LP problem and analyse the solution. Develop and solve a decision-making model – Testing whether learning outcomes 1 – 3 and 5 have been achieved.

Formative assessment(s) including online tests will be carried out during the semester to test students’ knowledge with regards to modelling and analysis of business problems.

**Assessment criteria**

To pass the module, students must achieve the 40% pass mark overall and the threshold of

30% in each assessment.

All assessments will be judged on the following criteria:

* + Ability to appropriately formulate and solve a linear programme;
  + Ability to interpret the graphical and/or computer solution to a linear programme and carry out sensitivity analysis;
  + Ability to select and apply an appropriate forecasting technique to solve a business problem;
  + Ability to evaluate and compare forecasting techniques using measures of forecast accuracy;
  + Demonstrate an awareness of common simulation methods and recognise their characteristics;
  + Ability to build a simulation model and design and compare various scenarios;
  + Ability to select and use an appropriate decision-making method to identify the best decision alternative for a business problem;
  + Level of communication skill demonstrated in synthesising and reporting solutions and recommendations to a business problem.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| ***Individual Coursework*** | *50* | *30* |  | *Coursework* |
| ***Exam*** | *50* | *30* |  | *Examination – Closed book* |

**Synoptic assessment**

This module draws on analytical skills learned elsewhere in the course. This module therefore engenders synoptic learning and its assessment is inherently synoptic.

**Sources**

**Essential reading**

Anderson, D.R., Sweeney, D.J., Williams, T.A. and Wisniewski M. (2014). *Introduction to Management Science: Quantitative Approaches to Decision Making*., 2nd Edition, Andover: Cengage Learning. (Online resource)

Oakshott, L. (2016). *Essential Quantitative Methods for business, management and finance*. 6th Edition, London: Palgrave. (Online resource)

**Further reading**

Camm, J.D. (2018). *Essentials of business analytics*. 3rd Edition, Australia: South-Western.

Winston, W.L. (2014). *Microsoft Excel 2013: data analysis and business modeling*. Microsoft Press. (Online resource)

Albright, S.C. and Winston, W.L. (2015). *Business analytics: data analysis and decision making*. 5th Edition, Stamford, CT, USA: Cengage Learning. (Online resource)

Albright, S.C. and Winston, W.L. (2012). *Management science modelling*. 4th Edition, Mason,

Ohio; Andover: South-Western; Cengage Learning.

Min, H. (nd). *Global business analytics models: concepts and applications in predictive, healthcare, supply chain, and finance analytics*. Pearson FT Press. (Online resource)

**WWW references**

*Analytics magazine (https://pubsonline.informs.org/magazine/analytics)*

**Link to the online reading list**

<https://rl.talis.com/3/westminster/lists/E1868275-3E00-EC25-94B0-2B18749FD5F6.html>

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Client- Server Architectures | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Cassim Farook | | |
| Extension: | Email: | |
| Host course and course leader: BSc Computer Science | | |
| Status: Core - BSc Computer Science | | |
| Subject Board: COMENG | | |
| Pre-requisites: None | Co-requisites: None | |
| Study abroad: Yes, but only provided that remote site can provide internet connection, the  proper development environment and remote invigilation. | | |
| Special features: None | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an  arrangement with an organisation(s) other than the University of Westminster. No | | |
| Summary of module content  This module introduces the concepts of the Client/Server and, more generally, Distributed Architecture that are at the base of systems where the constituting services can be virtualized, replicated and moved. The module explains two fundamental theoretical concepts: the implications of the transition from a single to a distributed execution space and that of digital transmission of data. The module also covers the fundamental aspects of data transmission. The Client/Server Paradigm is analysed in detail both as a simple example of a Distributed System and as a possible building block of more complex Distributed Architectures. The module also covers three main technologies used for implementation: sockets and Web Services (both SOAP and REST). | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Explain the fundamental concepts underlying the Execution Space, Digital Data Transmission (Protocol Stacks) and general concepts of Distributed Systems, explain the various aspects that occur when a program executed in a single execution space is re-designed as a distributed system;

LO2 Design and analyse simple networks and the protocols they use, Assess network

systems and technologies for suitability in various situations;

LO3 Choose between alternative paradigms and technologies for solving problems in distributed systems; Explain message passing and how it can be used in distributed systems;

LO4 Analyse and compare client-server tools and solutions and select the tools and architecture most suitable to implement a specified application. Design and implement web services-based client-server applications. Learn how to use development tools (NetBeans and Jax-WS) for the implementation, testing and executing of Web Services.

LO5 Explain how technologies such as Virtualization are used to implement and maintain Distributed Systems.

**Course outcomes the module contributes to**

BSc Computer Science L5.1, L5.2, L5.3

**Indicative syllabus content**

* + The concepts related to the **execution space (programs, threads, processes and variables)** and how these concepts related have to be re-thought in a distributed computational environment.
  + Fundamental aspects of **digital data transmission and networking** explaining the

**TCP/IP stack and covering the most commonly used protocols used**.

* + **The Theory of Distributed Systems**, in particular the **Client/Server Paradigm** both as a simple example of a Distributed System and as a possible building block of more complex Distributed Architectures.
  + **Two, Three and Multi-Tiered Systems**
  + The consequences of the distribution of the execution space across different elements (invocation of remote methods, marshalling of parameters, value consistency across different instances)
  + The **main implementation technologies** currently used for implementation: e.g. sockets (used mainly as an example of the implementation at low level) and Web Services (both SOAP and REST). Students are expected to learn at least one of these technologies and get acquainted with the tools needed for their development (e.g. NetBeans and Glassfish).
  + **How real Distributed Systems are managed**. Server Management, Virtual Machines and Containers are introduced and the use of these technologies for the implementation and maintenance of Distributed System is highlighted.

**Teaching and learning methods**

The module delivery is by a combination of lecture and laboratory work. Students attend a lecture (2 hours) and a practical tutorial (2 hours). The lectures will concentrate on theoretical aspects, and the tutorials focus on developing the students’ expertise in Internet and distributed systems programming. The tutorials will be articulated into three main sections:

1. Java recapitulation. A dummy prototype (to be executed within a single execution space) will be developed and a list of fundamental java concepts needed to pass the

coding exercise at the end of the Semester will be given as Formative Assessment. Students will have then a few months to address weaknesses (if any) in their Java knowledge

1. Web Service design and implementation and preparation for the Coding Exercises at the end of the Semester. Examples of increasing complexity of Client and Server will be developed and explained step by step during Tutorials. In the last tutorials, students that are interested in the development of a more complex Distributed System will be able to discuss a mini project with the tutor. This will be entirely voluntarily, students who prefer to practice for the Coding Exercise will be given tests for formative assessments.
2. All Lectures and Tutorials will be recorded on Panopto and made available on Blackboard.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24 |
| Seminar |  |  |
| Tutorial | Scheduled | 24 |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled |  |
| Supervised time in studio/workshop | Scheduled |  |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled |  |
| **Total Scheduled** |  | **48** |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | **200** |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

To pass the module students must be able to demonstrate that they have achieved a reasonable level of knowledge of the Learning Objectives. Two assessments (In Class Test – ICT) and Coding Exercise (CEX) will be used to assess the preparation of the students.

The rationale of the In Class Test (ICT) component is to evaluate the theoretical understanding of the concepts expressed in LO1 (execution space), L02 (networking and protocols), LO3 (Distributed System paradigms), LO4 (Web Services), and LO5 (Virtualization).

The rationale of the Coding Exercise Component (CEX) is to evaluate the practical understanding of the implementation of a simple prototype of a Distributed System (L01) which consists in a Client/Server system developed with Web Service tools and techniques (L04).

**Assessment criteria**

*Formative Assessment*

Formative assessments will be embedded within the teaching and learning activities of the

module. At every step of the development process, students will be required to regularly

demonstrate the extent of the development of their prototype to the rest of the class so that to get feedback from their peers but also from the members of the module team. This will result in constructive feedback which they can then re-invest in their development.

The first two or three Tutorials will be used first to assess the overall competence in Java of the students to urge those who have weakness in Java to address them as soon as possible.

The following tutorials focus on the core steps to develop a Client/Server system.

The last tutorials will be used to prepare the students to the CEX coding exercise and to provide Formative Assessments on their preparation.

During the last tutorials, the second part will be devoted in showing how a complex Distributed

Computational System is designed and developed.

All Formative Assessment (apart from the preparation of the CEX exam) can be provided on an individual or group basis depending on the student’s inclination.

*Summative Assessment*

Summative Assessment is divided into two separate tests: A Theory-Oriented In Class Test (ICT) and an In Class Test Coding Exercise (CEX)

The ICT will consist of 20 multiple choice and multiple answer questions about L01, L02, L03

and L05 and to pass it students will have to achieve a grade of 30.

The CEX will consist of a coding exercise whereby students will have to develop a Client/Server System. The assessment criteria is the following:

* + Be able to design, correctly implement and test a simple connection: 30
  + Be able to design, correctly implement and test a simple method (only simple types

passed and standalone logic of the method): 10

* + Be able to design, correctly implement and test an exception handling system for the

simple method: 10

* + Be able to design, correctly implement and test a small set of correlated methods that cooperate in implementing a task (e.g. one method that stores numbers on the server and one method that returns the average of all the added numbers): 20
  + Be able to design, correctly implement and test a small set of correlated methods different from any of the examples that have been used in the tutorials. The building blocks have been explained but the solution must show a certain degree of initiative and originality: 30

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| *In class test* | *40* | *30* |  | *Examination – closed book* |
| *Coursework* | *60* | *30* |  | *In Class Coursework – Closed Book* |

**Synoptic assessment**

This module draws on research, analytical, design and development skills learned elsewhere in the course. This module therefore engenders synoptic learning and its assessment is inherently synoptic.

**Sources**

Web Service Tutorial: <https://www.tutorialspoint.com/webservices/index.htm> SOAP tutorial: <https://www.tutorialspoint.com/soap/index.htm>

Jax-WS in NetBeans Tutorial: <https://netbeans.org/kb/docs/websvc/jax-ws.html>

**Essential reading**

Coulouris, G. Dollimore, J. Kindberg, T. and Blair, G. (2011) *Distributed Systems: Concepts and Design*, 5th Edition, Addison Wesley.

Tanenbaum, AS. and Wetherall, DJ. (2010), *Computer Networks,* 5th Edition. Prentice-Hall.

**Additional reading**

Programming Web services with SOAP, (2001) J. Snell, D. Tidwell, P. Kulchenko, O’Reilly Richardson, L., Amundsen, M., and Ruby, S. (2013), *RESTful Web APIs*, O'Reilly.

**Link to the online reading list**

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Database Systems | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Kaneeka Vidanage | | |
| Extension: | Email: | |
| Host course and course leader: BSc Computer Science | | |
| Status: Core - BSc Computer Science, BEng Software Engineering, BSc Business  Information Systems, BSc Data Science & Analytics; Option – BSc Computer Network Security. | | |
| Subject Board: COMENG | | |
| Pre-requisites: None | Co-requisites: None | |
| Study abroad: An alternative assessment can be produced to evaluate whether Study Abroad Students have passed the Learning Outcomes. | | |
| Special features: None | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an  arrangement with an organisation(s) other than the University of Westminster. No | | |
| Summary of module content  This module provides solid knowledge and skills in the area of database systems, SQL and XML. It covers the logical design of a relational schema. It also covers the implementation of the database in a major DBMS and the manipulation of the data using SQL. Subsequently, it considers the transformation and rendering of XML documents using XSLT and the extraction of elements from XML documents using XPath and XQuery. Finally, it explores issues related to NoSQL databases and XML databases. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Design a robust relational database schema using UML notations;

LO2 Produce robust SQL statements to create logically connected database tables and populate them;

LO3 Produce solid SQL queries to retrieve, aggregate, update and delete data from one or multiple database tables;

LO4 Transform and render XML documents using XSLT and address parts of XML documents using X-Path and XQuery;

LO5 Critically evaluate the needs for non-relational databases e.g. No-SQL databases and

XML Databases.

**Course outcomes the module contributes to**

BSc Computer Science L5.1, L5.2, L5.4

BEng Software Engineering L5.4, L5.7

BSc Business Information Systems L5.1, L5.2, L5.3, L5.4, L5.6

BSc Data Science & Analytics. L5.3, L5.4, L5.6

**Indicative syllabus content**

* + Logical database design to produce a relational schema: multiplicities, candidate keys, primary keys, foreign keys, composite keys, surrogate keys, entity integrity, relational integrity, general constraints.
  + Data definition in SQL to create and modify tables: data types, create, alter, drop,

rename and truncate.

* + Data manipulation in SQL to insert records into tables: insert into.
  + Data manipulation in SQL to retrieve, update and delete information from one or more tables: select, update, delete, aggregation, subqueries, joins.
  + Transformation and rendering of XML documents using XSLT.
  + Identification and extraction of elements and attributes from XML documents using X-

Path and XQuery.

* + Introduction of the non-SQL paradigm: NoSQL databases, document-oriented databases, XML databases.

**Teaching and learning methods**

The module delivery is supported by a combination of interactive lectures, computer lab- based practical exercises and self-study activities. Lectures will be used to present and critically evaluate the theoretical aspects of the course: logical relational schema modelling, alternative database paradigms, data definition and data manipulation in SQL, transformation and rendering of XML documents using XSLT and extraction of elements from XML documents using XPath and XQuery. In-lab practical Tutorials will be used to work on hands-on exercises that underpin the theoretical material to allow the students to construct solid technical skills in Entity Relational modelling, SQL, XML, XSLT, Xpath and Xquery. Roleplays simulating the interactions between a client and the database architect will be used throughout the module, particularly to support the formative component of the assessment.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 24 |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled |  |
| Supervised time in studio/workshop | Scheduled |  |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled |  |
| **Total Scheduled** |  | **48** |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | **200** |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

The summative assessment strategy first involves an individual coursework that will require for the students to demonstrate the knowledge and practical skills they will have acquired with regards to the design and development of a relational database application and their critical understanding of available database paradigms that can differ from the relational approach. Second, a portfolio of in-tutorial tests will be relied upon to assess the extent to which the students construct high quality technical skills in SQL and XML.

The individual coursework allows the student to demonstrate their ability to design a database, setting up this database on a RDBMS and querying it. The starting point could be a given project brief; the students will have to produce a sound relational schema for the database then implement, populate and query the corresponding database tables using SQL. The coursework also enables the students to develop an in-depth critical evaluation of the relational approach to database management versus the non-relational paradigm. Therefore, the coursework component addresses LO1, LO2, LO3 and LO5.

The In-tutorial tests will assess the students’ technical abilities and know-hows in the areas of SQL, XML, XSLT, XPath and XQuery. They could take the form of a series of 4 or 5 tests scheduled to take place during the tutorials, on a regular basis, to maximise the students’ engagement with the practical aspects of the module. SQL queries would be provided and would need to be evaluated, snippets of XML documents could be transformed and rendered using XSLT and a number of elements and attributes from XML documents could be identified and extracted using XPath and XQuery. This series of tests would thus cover LO2, LO3 and LO4

Formative assessment will be embedded within the teaching, learning and assessment activities of the module. The students will be required to demonstrate each phase of the design and development of the database and of the setting up of the database system and formative feedback will be continuously provided. In addition, the students will be asked to submit a work-in-progress version of their design and development products for the coursework so that they can be given targeted feedback during a dedicated week (Feedback Week) which they can take into consideration to produce the final version. Finally, the

formative assessment philosophy naturally drives the in-tutorial tests as the students will get immediate feedback on their performance and will be able to reinvest what they have learnt through this exercise into their subsequent learning in the rest of the module.

**Assessment criteria**

In order to pass the module, the students will have to demonstrate a detailed understanding of the design, development and application of database systems and of the different paradigms beyond the relational approach.

The criteria used to determine students’ performance will depend on the degree they are able to:

* + Produce a conceptual data model based on a project brief.
  + Convert a conceptual data model into a relational schema.
  + Write SQL queries to create the tables and query them.
  + Transform and render an XML document using XSLT.
  + Identify and extract elements from an XML document using XPath and XQuery.
  + Discuss and critically evaluate relevant database paradigms.

Meeting the above criteria will mean a pass mark. Better students are expected to be able to provide fuller answers; they should be able to demonstrate an insight into the topics covered and to produce innovative answers to more challenging issues and demonstrate higher levels of technical skills.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| ***Coursework*** | *60* | *30* |  | *Practical coursework– Report, UML diagrams and code* |
| ***In-Class Tests*** | *40* | *30* |  | *Written exam.* |

**Synoptic assessment**

This module draws on research, analytical, design and development skills learned elsewhere in the course. This module therefore engenders synoptic learning and its assessment is inherently synoptic.

**Sources**

## Essential reading

*Connolly, Begg & Strachan: Database Systems – A Practical Approach to Design, Implementation and Management. Pearson Education, 6th Edition, 2015. ISBN 978- 1292061184*

*Elmarsi & Navathe: Fundamentals of Database Systems. Pearson Education, 7th Edition, 2016. ISBN 978-1292097619*

*Sullivan: NoSQL for Mere Mortals. Addison-Wesley, 1st Edition, 2015. ISBN 978- 0134023212*

**Link to the online reading list**

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Game Engine Architecture | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Pumudu Fernando | | |
| Extension: | Email: | |
| Host course and course leader: BSc Computer Games Development | | |
| Status: Core - BSc Computer Games Development; Option - BSc Computer Science | | |
| Subject Board: COMENG | | |
| Pre-requisites: None | Co-requisites: None | |
| Study abroad: None | | |
| Special features: None | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an arrangement with an organisation(s) other than the University of Westminster: No | | |
| Summary of module content:  This module introduces students to modern game engine architecture and technologies. The conceptual architecture framework and the subsystem integration including the low- level foundation systems, the rendering engine, game asset management, the physics simulation, event-based gameplay system will be critically accessed. Students will gain the theory underlying the various subsystems that comprise a commercial game engine and the data structures and essential algorithms and develop practical skills that are typically used to implement a 3D game prototype using industry game engines. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Demonstrate a theoretical understanding of game engine architecture and essential algorithms and data structures;

LO2 Examine and evaluate the relevant technical issues and core functionalities of game engines and subsystem integration;

LO3 Use a professional game engine to design and implement 3D game prototypes;

LO4 Manage game assets from different sources to be used in the development process;

LO5 Examine and critically assess current problems and appreciate some of solutions available;

LO6 Communicate and present ideas by oral, visual and written means.

**Course outcomes the module contributes to:**

BSc Computer Games Development L5.1, L5.2, L5.3 , L5.6, L5.8 , L5.9

**Indicative syllabus content**

**Overview of game engines and trends**

* + game engine concept and historical development; game middleware and different commercial game engines review; game engine design approaches and technical requirements, and cross platforms issues; game engine future trends, the impact of Extended Reality (XR) on game experience

**Game engine architecture and subsystem integrations**

* + game engine conceptual framework and architecture, core system functionalities; game engine subsystems analysis including low-level foundation systems, rendering engine, sound engine, physics engine, resources and input manager; even-based game loop and scripting game logics; critical analysis of subsystem integrations and current problems and solutions

**Essential applied algorithms and technical issues**

* + specific technical issues and their requirements in game engines including low-level renderer, physics, animation, and sound; real time rendering pipeline and fast rendering algorithms (Binary Space Partition algorithm), dynamic level of details, and visibility testing in game engines; terrain generation methods and related algorithms; lighting techniques applied in game engine including illumination model, image-based lighting, ray tracing and radiosity

**Game Production and Game assets management**

* + modern game production workflow, integrated visual development environments; game assets management, pre-conditioning issues, dynamic scene graph and data structures; practical texture mapping methods applied in game engines; processing interactive objects and gameplay using visual programming; transferring reality with emerging media (VR and AR) to enhance user experience; raising professional awareness by add extra values e.g. social, educational, global, economic values in the design and development

**Teaching and learning methods**

Students attend one 2-hour lecture and one 2-hour tutorial each week. The lecture focuses on the selected topics within the syllabus areas in order to introduce the relevant theories, principles, and algorithms applied in game engines and to explain the integrated component- based game engine architecture and core subsystem functionalities that the student should

explore during tutorials. The tutorial concentrates on practical exercises on the subject knowledge and techniques learnt in the lectures and developing a 3D game prototype using industry game engines.

The module employs research-led pedagogic approaches to fostering students’ creativity through student engagement activities (e.g. workshops, group discussions) scheduled in this module that facilitate students to raise professional awareness and add extra values e.g. social, educational, global, economic values in the design and development by adopting an interdisciplinary approach. The technology enhanced learning approach has been adopted to blend social learning into formal learning. For example, using social media tools such as YouTube for student work demonstrations and assessment.

Furthermore, teaching and learning are informed and enriched by the research and scholar activities from the module team in wider subject areas including computer graphics, human computer interaction, game development, design with narrative experience.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 20 |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled |  |
| Supervised time in studio/workshop | Scheduled | 4 |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work based learning | Scheduled |  |
| **Total Scheduled** |  | **48** |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | **200** |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

The module will be assessed by two components that exam students in both theoretical knowledge and practical skills:

Assessment 1: One in-class test weighted at 30%. The in-class test shall assess the knowledge and understanding of the student in the required theoretical understanding of game engine architecture and essential algorithms and data structures. The assessment shall allow the student to demonstrate both knowledge and understanding of the game engine architecture. Through critical analysis of subsystem integrations and current problems and solutions, the student shall examine and evaluate the relevant technical issues and core functionalities of game engines and subsystem integration.

Assessment 2: A coursework – practical design and development work at 70%. The students shall demonstrate achievement of the learning outcomes by applying subject knowledge and essential technical issues through practical exercises and further demonstrate comprehensive practical skills and techniques to design and implement a 3D game prototype through examining current problems and appreciate some of solutions available.

Formative tutorial exercises will be made available on Blackboard for students to reinforce and practise their knowledge and learning and assess progress. These exercises will be made available on a week–by–week basis so that students can do them on a week–by– week basis too. Typically, formative assessment will be implemented by giving students mock online tests, surgery sessions, in-class discussions, Q&A sessions that provide feedback on their strengths and weaknesses in preparation for the later summative assessments.

**Assessment criteria**

Assessment 1 will be an in-class test on theory, principles, algorithms, data structures applied in the subject area. It will assess LO1, LO2, LO5. Assessment 2 will be the practical design and development of a 3D game prototype using a professional game engine. It will assess LO3, LO4, LO5 and LO6. A viva and demonstration will be part of component of the coursework assessment.

A pass level of achievement will be the proper understanding of the subject knowledge and the demonstration of the essential skills and techniques for practical implementation. Higher grades of achievement will require students to demonstrate solutions to a set of technical challenges in the subject. In addition to these summative assessments, several formative tasks are provided through supervised practical exercises and workshops, allowing the student to apply the topics discussed in the lectures. The students are thereby prepared for the in-class tests, as well as aid their understanding of the subject. The Coursework will assess the students on the learning outcomes for the design and development of a 3D game prototype to demonstrate a comprehensive understanding of modern game development approach using a commercial game engine.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| ***In-class Test*** | *30* | *30* |  | *Written exam* |
| ***Coursework: Practical Work*** | *70* | *30* |  | *Practical Development and viva* |

**Synoptic assessment**

There are no explicit synoptic assessments in this module.

**Sources Essential reading**

Jason Gregory (2014), *Game Engine Architecture (Second edition),* A K Peters/CRC Press. Andrew Finch (2014), *Unreal Game Engine,* 3DTotal Publishing.

**Unreal Resources and Documentation**

https://docs.unrealengine.com/en-US/index.html

**Further reading**

IEEE Computer Graphics & Applications (CG&A) <http://www.computer.org/cga/> IEEE Transaction on Visualization and Graphics

ACM SIGGRAPH Computer Graphics

**Link to the online reading list**

The reading list can be found on the module’s Blackboard site

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Human Computer Interaction & User Experience (HCI & UX) | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering School of Computer Science and Engineering | | |
| Module Leader(s): Theja Perera | | |
| Extension: | Email: | |
| Host course and course leader: BSc Computer Science | | |
| Status: Option - BSc Computer Science | | |
| Subject Board: COMENG | | |
| Pre-requisites: None | Co-requisites: None | |
| Study abroad: None | | |
| Special features: None | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an  arrangement with an organisation(s) other than the University of Westminster: No | | |
| Summary of module content  The module introduces students to the theoretical aspects of Human Computer Interaction (HCI) and User Experience (UX) as well as provides a practical understanding of current principles of effective interface design. The module equips students with the practical skills needed for the design and evaluation of interactive systems using a user-centred approach. It provides an understanding of the relevance and application of human abilities and limitations to the design of interactive systems and how context influences the human system interaction. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Explain the importance of understanding users and their cognitive aspects and how this knowledge can be applied to interface design, identify user needs and establish user requirements for different application domains;

LO2 Apply theory, design principles, practices and tools for the design, prototyping and evaluation of a user interface (UI);

LO3 Demonstrate knowledge of various guidelines and techniques applied in the process of interaction design;

LO4 Critically evaluate the usability and the user experience of various applications, systems

and products;

LO5 Plan and conduct user study/user research to inform development of systems and applications and appraise/recognise the ethical and professional issues involved.

**Course outcomes the module contributes to**

N/A as Module is not core

**Indicative syllabus content Introduction to core concepts**

Defining Human Computer Interaction, Human Factors, User Experience, Usability, User- centred Design/Human centred Design, Accessibility.

**The process of Interaction Design**

The four basic activities for interaction design and how interaction design activities can be integrated into other development lifecycles. How the human-centred design approach to interactive systems development aims to make systems usable and useful.

**Interaction Design in practice**

Agile UX and Lean UX, Design Patterns, Interaction Design tools (low fidelity/paper-based prototyping, high fidelity prototypes), Design Principles and Guidelines, Conceptual Design, Generating prototypes, Accessibility.

**Cognitive aspects**

How cognitive processes (i.e. attention, perception, memory, learning, reading, speaking, and listening, problem-solving, planning, reasoning, and decision-making) affects the human computer interaction and respective design implications.

How understanding users and the knowledge of human capabilities and limitations can inform the design of technologies that can extend human capabilities and compensate for human weaknesses.

**Interfaces**

Interface types (for example, Command, Graphical, Web, Mobile, Virtual Reality, Appliance, Voice, Touch, Gesture, Haptic, Multimodal, Augmented Reality, Wearables, Ubiquitous Computing, Emerging User Interfaces, etc. Emotional Interaction (emotional design, affective computing, persuasive technologies and behavioural change, anthroporphism)).

**Data gathering, analysis, interpretation**

Techniques for data gathering (choosing and combining techniques), difference between qualitative and quantitative data and analysis, interpreting and presenting findings. Ethical and professional issues involved.

**User centred Design/Human-centred Design and Requirements gathering**

Data gathering for requirements, personas and scenarios, etc.

**Evaluation**

Types and methods of evaluation - for example, usability testing, conducting experiments, field studies, heuristic evaluation and walk-throughs, analytics and A/B Testing, predictive models, etc.

**Teaching and learning methods**

This is a semester-based module. This module will be delivered using a combination of lectures and tutorials on a weekly basis. In the lecture (2h) students will be introduced to the theoretical aspects and concepts of Human Computer Interaction (HCI) and User Experience (UX) as well as offered a practical understanding of current principles of effective interface design. In the tutorial (2h) students will be challenged to apply the concepts learned through the lecture in a variety of settings in order to achieve the LOs and consequently become equipped with the practical skills needed for the design and evaluation of interactive systems using a user-centred approach. Students will have the opportunity to work in small groups, apply the knowledge acquired during the lectures through practical, problem solving exercises, case studies, active learning and interaction with others. In addition, the tutorials will introduce an industry-standard tool, used for the development of prototypes/UX deliverables.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24h |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 24h |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled |  |
| Supervised time in studio/workshop | Scheduled |  |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled |  |
| **Total Scheduled** |  | 48h |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152h |
| **Total student learning and teaching hours** |  | 200h |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

The assessment is designed to measure the full range of learning outcomes. It is sufficiently diverse as it offers more latitude for students to demonstrate their knowledge.

The Coursework will take students through the complete development lifecycle of an interactive system and the process of Interaction Design. The overall purpose of the Coursework is that students design (or extend) and assess (using a user-centred approach) an interactive system (such as an online system, mobile, smart systems, a voice user interface (VUI), etc.) following a human-centred design approach. This approach focuses on the users, their needs and requirements and applying human factors/ergonomics and usability knowledge and techniques with the aim to make systems usable and useful.

The Coursework is industry-relevant and aligned with the University's assessment strategy to increase employability. It allows students to build on and demonstrate knowledge and skills at progressive levels. Students completing successfully the Coursework will be able to demonstrate that they have achieved the respective learning outcomes of the module (LO1, LO2, LO3, LO5) as well as they can use this to demonstrate their UX competency to potential employers.

Formative Assessment

Students will study an environment or a situation and either identify a problem and suggest the development of a new interactive system to address this problem or identify a problem of a current system and suggest its redesign and/or extension in order to identify and address user needs and requirements. Students will submit their ideas/proposals of the system they plan to design or extend and will receive feedback that will guide and support them towards working and completing the Coursework. Informally, this will assess Learning Outcomes (LO1, LO4).

Summative Assessment

For the first part of the Coursework, students will work in small groups (for example, ideally in groups of four members). Having obtained formative feedback for the interactive system they propose to design or extend they will then identify user needs and establish requirements, design or extend the system effectively through a user-centred approach, communicate this design, perform evaluation and present their findings (deliverables would be focusing on presenting findings from user research and low fidelity prototyping). At this stage there might also be opportunities for peer-to peer feedback (among different groups).

It is often common industry practice that UX professionals working with other team members to create documentation and deliverables, especially during the user research phase. Therefore, it is important having students working in small groups in the first part of the Coursework where tasks are designed in such way that involve high levels of collaboration and negotiation between the members.

For the second part of the Coursework, students will work individually. Having taken into consideration their findings from the first part, they will then develop a high-fidelity prototype (using an industry standard tool or a software tool or language of their choice) while considering the interface design in detail. They will then proceed with planning and conducting evaluation of the prototype and discuss their findings. (deliverables would be focusing on high- fidelity prototyping and evaluation - details of the type of the evaluation will be provided). Having students working individually in this part of the Coursework offers a degree of freedom to students to choose the tool(s) they will use to implement the high-fidelity prototype (based on their preference and/or competence), reflect on the previous findings and take full responsibility on their design decisions as well as on how they will proceed with the evaluation.

The Examination will assess knowledge of core areas through the use of short-answer questions covering the range of approaches and topics included in the module (LO1, LO2, LO3, LO4, LO5).

**Assessment criteria**

The assessment evaluates whether and to what extent the student has demonstrated achievement of the learning outcomes.

The Coursework requires an understanding of what is an effective user interface, critically reviewing the needs of different user groups, conducting user research, suitably applying techniques and methodologies for the design and construction of low and hi-fidelity prototypes, applying evaluation methods and presenting findings. Marks will be allocated based on the extent the student has demonstrated the ability to achieve the above.

The Examination will assess knowledge of core areas through the use of short-answer questions covering the range of approaches and topics included in the module and marks will be allocated based on the extent the student has demonstrated such knowledge and understanding.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| Coursework | 60 | 30 |  | Portfolio |
| Examination | 40 | 30 |  | Closed Exam |

**Synoptic assessment**

None

**Sources**

**Essential reading:**

Sharp, H., Preece, J. Rogers, Y. (2019). *Interaction Design: Beyond Human Computer Interaction* (5th Edition). John Wiley & Sons Ltd, Chichester.

Benyon, D. (2019). *Designing User Experience: a guide to HCI, UX and interaction design*. (4th Edition). Pearson UK.

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: strategies for effective human-computer interaction*. (6th Edition). Pearson.

Gothelf, J., & Seiden, J. (2016). *Lean UX: designing great products with agile teams* (2nd

Edition). O'Reilly Media, Inc.

**Further reading:**

Lazar, J., Feng, J. H., & Hochheiser, H. (2017). *Research methods in human-computer interaction*. (2nd Edition). Morgan Kaufmann.

Travis, D., & Hodgson, P. (2019). *Think Like a UX Researcher: How to Observe Users, Influence Design, and Shape Business Strategy*. CRC Press.

**Periodical references:**

Interactions (ACM) - [http://interactions.acm.org](http://interactions.acm.org/)

**Additional references:**

Interaction Design Foundation - [http://www.interaction-design.org](http://www.interaction-design.org/)

ACM Special Interest Group on Computer Human Interaction - [http://www.sigchi.org](http://www.sigchi.org/) Proceeding of ACM Conference on Human Factors in Computing Systems - <http://www.sigchi.org/conferences>

Proceeding of BCS Conference on HCI - [http://www.bcs-hci.org.uk](http://www.bcs-hci.org.uk/)

UX & Usability Articles from Nielsen Norman Group - https://[www.nngroup.com/articles](http://www.nngroup.com/articles)

**Link to the online reading list**

The link will be provided (and it will replace the reading list below)

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Information Technology Security | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Austen Mascaranghe | | |
| Extension: | Email: | |
| Host course and course leader: BSc Business Information Systems | | |
| Status: Core - BSc Business Information Systems; Option – BSc Computer Science, BSc Data Science and Analytics; Westminster Plus Elective | | |
| Subject Board: COMENG | | |
| Pre-requisites: None | Co-requisites: None | |
| Study abroad: A case-study analysis will be used for assessment 2 | | |
| Special features: None | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an arrangement with an organisation(s) other than the University of Westminster. No | | |
| Summary of module content  This module examines the issues involved with recognising security threats to computer systems, their consequences and methods of dealing with such threats. In particular, it provides an overview of access controls, software development security, business continuity, legal issues and compliance, and physical security. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Describe and discuss the major areas of threat to security across the range of systems and the breadth and limitations of related controls;

LO2 Identify and assess information security risks and propose systems and controls; LO3 Describe and discuss the legal and regulatory consequences of security threats;

LO4 Describe software development security and evaluate risk in software design, and

select and evaluate different approaches to business continuity and disaster recovery; LO5 Analyse issues and contrast solutions relating to environmental security.

**Course outcomes the module contributes to**

BSc Business Information Systems L5.1, L5.2, L5.3, L5.4, L5.5, L5.6

**Indicative syllabus content**

* + Access Control – examples include Concepts / Methodologies / Techniques, Effectiveness, Attacks.
  + Information Security Governance and Risk Management – examples include Security Governance and Policy, Information Classification/Ownership, Contractual Agreements and Procurement Processes, Risk Management Concepts, Personnel Security, Security Education, Training and Awareness, Certification and Accreditation.
  + Software Development Security – examples include Systems Development Life Cycle (SDLC), Application Environment and Security Controls, Effectiveness of Application Security.
  + Business Continuity and Disaster Recovery Planning – examples include Business

Impact Analysis, Recovery Strategy, Disaster Recovery Process, Providing Training.

* + Legal, Regulations, Investigations and Compliance – examples include Legal issues, Investigations, Forensic procedures, Compliance Requirements/Procedures.
  + Physical (Environmental) Security – examples include Site/Facility Design

Considerations, Perimeter Security, Internal Security, Facilities Security.

* + Brief introduction (from a people-focus) of Telecommunications and Network

Security, Cryptography, Security Architecture and Design, Operations Security.

**Teaching and learning methods**

The module delivery is by a combination of interactive lectures, lab-based practical exercises and in-class debates. Interactive lectures will include, as appropriate to the background of the class, Q&A sessions and elements of flipped classroom, with the exception of the very technical elements. In-class quizzes will be used, where appropriate, to break sessions and engage students in upcoming topics. Conflicts between theory and industrial practice, and debates on ethical dilemmas and cultural perspectives will be threaded across the lectures and include, where relevant, students’ own experiences as users of technology, through open class discussions. Research and evaluation of sources in the subject area will be approached from an applied perspective and students will be guided with practical activities and peer review to improve their critical assessment of on-line sources. Reference material and references will be available on the web. Guidance will be given for some of the self- managed independent study as appropriate for level 5.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 24 |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled |  |

|  |  |  |
| --- | --- | --- |
| Supervised time in studio/workshop | Scheduled |  |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled |  |
| **Total Scheduled** |  | **48** |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | **200** |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

LO1, LO4 will be assessed through a template-based report on a given case-study exploring multiple technical aspects; this will involve group-work to benefit the development of thoughts, supporting arguments and awareness of limitations, with evidence of peer-review within the group. Where class-sizes permit, each student will have a unique coursework focus to reduce opportunities of assessment offences.

LO2, LO3, LO5 will be assessed by formal examination requiring students to demonstrate knowledge of established concepts and apply them by selecting appropriate solutions to given scenarios under time constraint.

Formative assessment: Student learning will be supported – and students will be able to gauge their progress towards the learning outcomes – through on-line activities as well as peer-assessed analysis of industry scenarios complemented by staff feedback. Students will also have set deadlines for formative exercises, typically for discussion/review in tutorial time, including on research and evaluation of information sources, followed by in-class staff feedback.

**Assessment criteria**

For a pass mark, students will be expected to demonstrate their understanding of the security issues discussed in their given scenario, undertake research beyond sources given by the module team, and communicate with appropriate format and originality.

In addition, for a 2:2, students will be expected to show reflective thinking and some understanding of the implications of their proposed solutions in their given scenario.

In addition, for a 2:1, students will be expected to evaluate their sources and be reflective across their work, and present some well-constructed and presented arguments in their given scenario.

In addition, for a 1st, students will be expected to demonstrate a degree of criticality in their use of external sources and to synthesise solutions with some respect to how they impact each other within the wider context of their given scenario, with the degree of originality and quality of communication leading to the higher end of 1st class marks.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g.**  **essay, presentation, open exam or closed** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | **exam)** |
| ***Group threat analysis and defence*** | *50* | *30* |  | *Template-based case- study report* |
| ***Exam*** | *50* | *30* |  | *Written exam - closed book* |

**Synoptic assessment**

This module draws on technical, analytical, research and presentation skills learned elsewhere in the course. This module therefore engenders synoptic learning and its assessment is inherently synoptic.

**Sources**

**Link to the online reading list**

https://rl.talis.com/3/westminster/lists/1952F0BB-E8A3-04D9-A71B-9D8947C8096F.html

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Machine Learning and Data Mining | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Fathima Farhath | | |
| Extension: | Email: | |
| Host course and course leader: BSc Data Science and Analytics | | |
| Status: Core – BSc Data Science and Analytics, BEng Software Engineering; Option – BSc Computer Science, BSc Computer Games Development, BEng Smart Computer Systems | | |
| Subject Board: COMENG | | |
| Pre-requisites: None | Co-requisites: None | |
| Study abroad: Not available | | |
| Special features: None | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an arrangement with an organisation(s) other than the University of Westminster. No | | |
| Summary of module content  This module provides an understanding and hands-on experience in the fields of machine learning and data mining, covering the full life-cycle from preparing data to validating and optimising the learned model. The module covers different algorithms and approaches to machine learning and data mining, and the issues of using them on data sets of different sizes and complexity. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Suitably prepare a realistic data set for data mining / machine learning and discuss issues affecting the scalability and usefulness of learning models from that set;

LO2 Effectively implement, apply and contrast unsupervised/supervised machine learning / data mining algorithms for simple data sets, discussing and addressing issues arising from their use, in work-based scenarios from industry and wider community;

LO3 Evaluate, validate and optimise learned models;

LO4 Effectively communicate models and output analysis in a variety of forms to specialist and non-specialist audiences;

LO5 Demonstrate an awareness of ethical issues and personal responsibility in the preparation of data and the implementation, application and communication of machine learning / data mining algorithms in a professional working environment

**Course outcomes the module contributes to:**

BSc Data Science and Analytics L5.3, L5.4, L5.5, L5.7, L5.8

BEng Software Engineering L5.1, L5.2

**Indicative syllabus content**

* Introduction to machine learning and data mining, and the ethical and privacy context

of their real-life application.

* Issues regarding suitability and pre-processing of data, e.g. data types, dimensionality reduction, entropy, noise, bias, overfitting.
* Supervised learning, e.g. Bayesian, Decision Trees, Neural Nets, Regression, SVM
* Unsupervised learning, e.g. Clustering, Neural Nets, Anomaly detection
* Other approaches to learning, e.g. Ensemble, Genetic Algorithms
* Issues of scalability and large data sets
* Methods for evaluation, validation and optimisation of models
* Industry applications of machine learning and data mining and emerging market trends

**Teaching and learning methods**

The module delivery is by a combination of lecture and lab-based tutorials. Lectures will include industry input and, where appropriate, exposure to talks from industry partners, while tutorials will be based on real-life applications of the subject area from industry and the wider community. Reference material and references will be available on Blackboard. Guidance will be given for some of the self-managed independent study as appropriate for level 5.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 22 |
| Workshops | Scheduled |  |
| **Total Scheduled** |  | 48 |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | 200 |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

Students will need to engage with the issues of machine learning algorithms from the start of the module, applying the theory from the lectures to hands-on tutorial work, informed by industry input, that will help them build their own learning models.

For their coursework students will demonstrate their understanding for assessment by using at least two learning algorithms from different learning paradigms, and show they can suitably prepare the data for these algorithms and evidence the effectiveness of their solution through appropriate validation/evaluation mechanisms. [Individual Coursework: Learning Outcomes 1, 3 and 4]

Students will also need, under time restricted conditions, to explain issues of different algorithms and identify and discuss problems inherent in different data sets used in industry for machine learning and data mining. The assessment will take place in a computer Lab where the students will be assessed on both theoretical and practical aspects. [Lab Controlled Assessment: Learning Outcomes 2 and 5]

Formative assessment: Student learning will be supported – and students will be able to gauge their progress towards the learning outcomes – through tutorial exercises and, as appropriate for the cohort needs, on-line tests as well as peer-supported analysis of industry scenarios complemented by staff feedback. In the first half of the module, students are typically expected to demonstrate to their tutor and peers how to use data appropriately and build models without the complexities of comparing algorithms or optimisation approaches. This will lay the foundations of knowledge and tools for the formative exercises of the second half of semester where students would normally be expected to build, discuss, evaluate and optimise their own learning systems during tutorials. Based on cohort needs, and in the context of the course employability policy, students are expected to also experience professional feedback through module-led industry reviews of selected top student submissions, within and after the end of the module.

**Assessment criteria**

The assessment criteria are derived from the Learning outcomes, and will be based on the extent to which a student demonstrates that the learning outcomes of the module have been achieved.

To pass the module, students must demonstrate an ability to:

* Use learning algorithms effectively on realistic data sets;
* Recognise the limitations and possible pitfalls of the proposed solutions;
* Evaluate, validate and optimise learned models;
* Identify and evaluate ethical issues related to the use of data and implementation of algorithms.
* Demonstrate a methodical approach to the machine learning/data mining life-cycle;
* Engage with the broader field of current and emerging technologies and applications.

In addition, for a 2:2, students will be expected to show reflective thinking and some understanding of the implications of their proposed solutions in their given scenario in an professional environment.

In addition, for a 2:1, students will be expected to evaluate their sources and be reflective across their work, and present some well-constructed and presented arguments supporting their decisions.

In addition, for a 1st, students will be expected to demonstrate a degree of criticality in the choices with some respect to how they impact each other within a wider professional context, with the degree of originality and quality of communication leading to the higher end of 1st class marks.

**Assessment methods and weightings***.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting %** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| **Coursework** | 40 | 30 |  | Individual Coursework |
| **Lab controlled assessment** | 60 | 30 |  | Lab exam |

**Synoptic assessment**

This module draws on technical, analytical and communication skills learned elsewhere in the course. This module therefore engenders synoptic learning and its assessment is inherently synoptic.

**Sources**

**Link to the online reading list**

https://rl.talis.com/3/westminster/lists/AF5BE045-CDC1-6E65-1FF3-9193CD72C88B.html

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Maths and Physics for Games | | |
| Module code: | Credit level: 5 | Length: 1 semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Dr. Randil Pushpananda | | |
| Extension: | Email: | |
| Host course and course leader: BSc Computer Games Development | | |
| Status: Status: Core – BSc Computer Games Development; Option - BSc Computer Science | | |
| Subject Board: COMENG | | |
| Pre-requisites: none | Co-requisites: none | |
| Study abroad: none | | |
| Special features: none | | |
| Access restrictions: none | | |
| Are the module learning outcomes delivered, assessed or supported through an arrangement with an organisation(s) other than the University of Westminster. No | | |
| This module covers the core physics and mathematical principles that are typically required for the creation of games or other scientific applications that implement realistic physical behaviour of objects. The module covers the necessary classical mechanical principles and mathematical techniques required to simulate the physics of games objects in a high-level language. It also relates this material to the utilisation of game-based physics engines. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Demonstrate an understanding of the physics required in a given application scenario in terms of forces and the laws of motion including conservation of momentum, gravity and friction;

LO2 Analyse a given kinetic problem and be able to apply the appropriate mathematical techniques in order to solve that problem;

LO3 Apply time-stepping numerical techniques to a given a game-based physics problem and be to implement a solution in a high-level programming language;

LO4 Critically apply their knowledge of vector maths and physics to a kinematic problem in a game engine;

LO5 Communicate and defend their work by both written and oral means.

**Course outcomes the module contributes to**

BSc Computer Games Development L5.1 ,L5.2

**Indicative syllabus content**

* Newton’s Laws, inertia, force, and units
* Integral and differential calculus
* Vector Mathematics and its application to kinetic problems
* Kinematics; linear, angular velocity, conservation of momentum
* Collision types, rigid bodies, and implementation of collision detection
* Projectiles and trajectory
* Introduction to Unity – creating simple scenes to test physics-based scenarios
* Working with game engines – understanding the features and limitations of physics engines
* Numerical methods – working with vector quantities in 2 and 3 dimensions to solve kinematic physics problem

**Teaching and learning methods**

Timetabled activity is divided equally between lectures and tutorials, consisting of 2 hours of each in a week. Lectures are used to introduce the relevant physical concepts and mathematics and also computational numerical approaches to solving kinematic physics problems. Mathematical problems shall be worked thorough by hand as well as several live code builds to demonstrate numerical methods and techniques shall be included as part of the lecture component remit. Visual example of Newton Laws are demonstrated using a typical games engine.

The tutorials will be based around formative problem-solving exercises that will reflect the materials learned in the lectures and incrementally build the student’s knowledge and skills. Additional Blackboard based short formative tests and reflection exercises shall be undertaken to help reinforce knowledge and to assess student progress. Tutorials shall include generation of solutions to mathematical problems and the codification of those solutions. Tutorial exercises shall be self-contained, and feedback will be given to the student during the tutorial session. Worked examples shall also be provided in tutorials so libraries of code samples and technique can be built and utilised to help develop the coursework.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 24 |

|  |  |  |
| --- | --- | --- |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled |  |
| Supervised time in studio/workshop | Scheduled |  |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled |  |
| **Total Scheduled** |  | 48 |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | 200 |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

The module will be assessed by one in-class test and one practical coursework. The in-class test shall assess the knowledge and understanding of the student in the required mathematical and physical concepts and shall also allow for diagnostic testing and feedback in preparation for the practical coursework. The coursework shall involve the design and implementation of an application in a high-level programming language that incorporates the mathematical/physical principles and techniques delivered on the module and assessed in the in-class test. Graphical assets (game objects) required for the coursework shall be provided by using a standard game engine so that the student can focus on applying the mathematics and physics to those assets. The coursework shall allow the student to demonstrate both knowledge and understanding of the taught physical concepts and mathematics and the ability to codify a given problem and apply numerical methods. By building a physics-based application from first principles the student shall have greater insight into the utilization of typical industry standard physics engines and shall also be able to apply these techniques to other scientific problems.

**Assessment criteria**

The student shall demonstrate achievement of the learning outcomes by being able to solve mathematical problems and be able to apply the mathematics and physics taught in an application and also through the in-class test. The in-class test shall verify the level of understanding of the concepts and the ability to solve relevant mathematical and physics- based problems and this knowledge is utilised in the coursework component of the assessment. For the coursework, higher marks are awarded for correctness and completeness and the ability to efficiently solve the given the given kinetic problems in terms of code efficiency and mathematical formulation. The student shall undergo a viva-voce for the practical coursework where they shall need to demonstrate their application and understanding and defend design decision.

In-class test: Solve vector and physics based mathematical problems in 2D and 3D, demonstrate an understanding of the laws of motion and conservation of momentum by solving simple 1D and 2D problems utilising the equations of motion. (LO1, LO2, LO3).

Coursework 2: Implementation of a simple physics engine and 2D/3D vector library to solve simple trajectory, rotation, and conservation of momentum problems using a standard numerical method. (LO3, LO4, LO5).

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| In-class test | *30* | *30* |  | *In-class test* |
| ***Coursework*** | *70* | *30* |  | *Coursework and Viva* |

**Synoptic assessment**

There are no explicit synoptic assessments in this module.

**Sources**

Beiser, A. (2012). Applied physics. New York: McGraw-Hill.

Millington, I. (2010). Game physics engine development: how to build a robust commercial- grade physics engine for your game 2nd ed. Amsterdam: Morgan Kaufmann Publishers.

Available from [http://www.sciencedirect.com.ezproxy.westminster.ac.uk/science/book/9780123819765.](http://www.sciencedirect.com.ezproxy.westminster.ac.uk/science/book/9780123819765)

Newton’s laws of motion - Wikipedia. Available from https://en.wikipedia.org/wiki/Newton%27s\_laws\_of\_motion.

Stroud, K.A. and Booth, D.J. (2013). Engineering mathematics 7th ed. Basingstoke: Palgrave Macmillan.

**Unity Resources**

Unity - Manual: Physics. Available from: https://docs.unity3d.com/Manual/PhysicsSection.html.

Unity - Manual: Unity User Manual (2019.2) . Available from: https://docs.unity3d.com/Manual/index.html.

Unity Training and Tutorials | Lynda.com). Available from: [https://w](http://www.lynda.com/Unity-training-tutorials/1242-0.html)ww[.l](http://www.lynda.com/Unity-training-tutorials/1242-0.html)y[nda.com/Unity-training-tutorials/1242-0.html](http://www.lynda.com/Unity-training-tutorials/1242-0.html)

**Link to the online reading list**

The reading list can be found on the module’s Blackboard site

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Mobile Application Development | | |
| Module code: | Credit level: 20 | Length:1 Semester |
| UK credit value: 20 | ECTS value: | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Torin Weerasinghe | | |
| Extension: | Email: | |
| Host course and course leader: BSc Computer Science | | |
| Status: Option - BSc Computer Science, BEng Software Engineering, BSc Computer Games Development | | |
| Subject Board: COMENG | | |
| Pre-requisites: none | Co-requisites: none | |
| Study abroad: N/A | | |
| Special features: N/A | | |
| Access restrictions: N/A | | |
| Are the module learning outcomes delivered, assessed or supported through an arrangement  with an organisation(s) other than the University of Westminster. No | | |
| Summary of module content  The module is an introduction to software development on mobile devices such as mobile phones, tablets and wearables. It concentrates on the Android platform. The main contents include: The Android mobile programming architecture. Restrictions of using small devices such as mobile phones tablets and wearables. Programming user interfaces, networking, persistent storage and multi-threading. Device profiling, application deployment and installation. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Apply mobile application programming tools and development environments to implement and deploy a mobile application based on a requirements document;

LO2 Explain in detail how to provide portability over a wide range of devices;

LO3 Describe and justify the different components which are needed to implement application specific functionality, such as network connectivity, data persistence, and location based services;

LO4 Discuss in detail the Android mobile application development cycle

LO5 Test and validate the functionality of a mobile application.

**Course outcomes the module contributes to**

N/A (the module is not core in any oft he courses offered)

**Indicative syllabus content**

* + Introduction to Android. Android Architecture. A "Hello Android" Application. Creating Layouts and Resources.
  + Event Handling. Text and Scrolling Views. Buttons and other Input Controls.
  + Activities and Intents
  + Activity Lifecycle. Implicit Intents.
  + The Android Studio Debugger. App Testing. The Android Support Library.
  + RecyclerView.
  + Data Storage 1: Working with Databases. SQLite. Room.
  + Background Tasks and Internet Connection. Connecting to the Network.
  + Data Storage 2: Shared Preferences and Settings.
  + Fragments.
  + Location and Maps
  + Usability for Android applications. Understanding user requirements and backgrounds; application and content goals; core usability design approaches; design elements; leveraging the device feature sets and using industry standard guidelines;

**Teaching and learning methods**

This module will be delivered in a mixture of presentations and lectures and supervised practical work. The taught material will be delivered in a 2- hour lecture and 2-hour tutorial mode for each teaching week. Outside formal class time there will be online support via Blackboard.

Elements of self-study will also be included in the module. Students will be required to self- study certain additional aspects of Android mobile programming and produce appropriate software solutions related to the material studied.

Extra exercises and tasks will be given to the students during the tutorials and students are expected to work towards their solution outside the formal tutorial slots. The produced solutions of these will be discussed with their module tutor, during the next tutorial.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled |  |
| Project supervisor | Scheduled |  |

|  |  |  |
| --- | --- | --- |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled | 24 |
| Supervised time in studio/workshop | Scheduled |  |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled |  |
| **Total Scheduled** |  | 48 |
|  |  |  |
| Placement | Placement | 110 |
| Independent study | Independent | 158 |
| **Total student learning and teaching hours** |  |  |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

The first programming assignment provides for a practical application of the concepts encountered in the lectures and first-hand experience of the full lifecycle applied to a medium scale mobile application development project, including use of the Eclipse development environment or Android Studio (LO1, LO4, LO5). The second coursework covers all the LOs.

**Assessment criteria**

A pass level would be demonstrated by an ability to produce a basic portable mobile application that can handle user input and output, navigate a basic set of menus and dialogues and produces some useful functionality to the user. To gain a pass level the student should also be able to demonstrate an understanding for the development of a basic mobile application. A higher level of ability would be indicated by the student being able to explain and demonstrate how to apply all the programming techniques shown such as persistent storage, network access and robustness.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| ***Coursework 1*** | *50* | *30* |  | *Software project* |
| ***Coursework 2*** | *50* | *30* |  | *Software project* |

**Synoptic assessment**

This module draws on technical and analytical skills learned elsewhere in the course. This

module therefore engenders synoptic learning and its assessment is inherently synoptic.

**Sources**

**Essential reading list**

1. *Google Android Developer Fundamentals 2.*

https://google-developer-training.github.io/android-developer-fundamentals-course- concepts-v2/index.html

1. Burnette Ed (2015) *Hello Android*, 4th edition. The Pragmatic Bookshelf

**Further Reading**

1. Bill Phillips, Chris Stewart, Brian Hardy and Kristin Marsicano, *Android Programming: The Big Nerd Ranch Guide*, (2017) Big Nerd Ranch, third edition, LLC.
2. Abelson W. Frank, Sen Robi, King Chris (2011), *Android in Action*, second edition,

Manning.Conder

1. *S*hane, Darcey Lauren (2010), *Android Wireless Application Development,* second

edition, Addison Wesley.

1. Ian G. Clifton (2015), *Android User Interface Design: Implementing Material Design for Developers*, second addition, Addison-Wesley Professional.

**Link to the online reading list**

**https://rl.talis.com/3/westminster/lists/CDBEBFFE-BF47-4E6C-EE9A- B60CD26A4FC5.html**

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Object Oriented Programming | | |
| Module code: 5 | Credit level: Level 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Guhanathan Poravi | | |
| Extension: | Email: | |
| Host course and course leader: BSc Computer Science | | |
| Status: Core - BSc Computer Science, BEng Software Engineering; Option - BSc Data Science & Analytics | | |
| Subject Board: COMENG | | |
| Pre-requisites: None | Co-requisites: None | |
| Study abroad: An alternative assessment can be produced to evaluate whether Study Abroad Students have passed the Learning Outcomes. | | |
| Special features: None | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an arrangement with an organisation(s) other than the University of Westminster. - No | | |
| Summary of module content  This module will teach the fundamental ideas behind the object-oriented approach to programming. It will provide students with knowledge and practical experience in writing computer programmes using object-oriented programming techniques. It will cover in a practical way the design and implementation of object-oriented software for software applications through the entire software development lifecycle. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Identify and justify good practices in object-oriented software development. Communicate the aspects of object-oriented programming which are advantageous when compared to non-object-oriented paradigms;

LO2 Acquired detailed knowledge of concepts of object-oriented programming and apply characteristics, tools and environments to adapt to new computational environments and programming languages which are based on object-oriented principles;

LO3 Design implement applications based on an object-oriented programming language, given a set of functional requirements. Use APIs which have not been exposed to previously, in order to develop an application requiring specialised functionality;

LO4 Design and Implement graphical interfaces using an object-oriented programming language;

LO5 Apply appropriate techniques for evaluation and testing and adapt the performance accordingly.

**Course outcomes the module contributes to**

BSc Computer Science L5.1 L5.3 L5.4 L5.5 L5.7 L5.8

BEng Software Engineering L5.1 L5.2 L5.3 L5.4 L5.5

**Indicative syllabus content**

* Object Oriented Programming principles and characteristics.
* Classes and Objects. The usage of APIs. Packages and namespaces.
* Polymorphism. Inheritance. Abstraction. Encapsulation. Access Specifiers. Abstract Classes and Interfaces. Overloading and overriding.
* How to design classes. UML design (UML class diagram and UML use case diagram)
* Graphical User Interfaces. Event Driven Design.
* Collections and generics. Data structures.
* Principles of concurrency and implementing threads.
* Program correctness: Defensive Programming (secure coding, exception handling, recovery), the concept of specification.
* Input/output and streams.
* Fundamental design patterns (examples could include but are not limited to: factory, MVC, singleton, adapter, proxy, facade, decorator).
* The role of algorithms in the problem-solving process. Performance Issues. Testing. Unit testing. Refactoring.

**Teaching and learning methods**

This module will be delivered in a mixture of presentations and lectures and supervised practical work. The taught material will be delivered in a 2-hour lecture and 2-hour tutorial mode for each teaching week. Technology-enhance learning, such us online quizzes and multiple-choice questionnaires will be used to promote students’ participation and engagement during classes. Outside formal class time, there will be online support via Blackboard: lectures will be recorded and made available to students, solutions to code exercises will be provided, Blackboard forum and Padlet will be used to open discussions with students and reply questions. The content of the module is highly practical with

sufficient emphasis on problem solving skills. These skills will be developed through a number of appropriate tutorial exercises extending the relevant theoretical material introduced in the lectures.

Elements of self-study will also be included in the module. Students will be required to self- study certain additional aspects of object-oriented programming and produce appropriate software solutions related to the material studied.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24 |
| Tutorial | Scheduled | 24 |
| **Total Scheduled** |  | 48 |
|  |  |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | 200 |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

The summative assessment strategy involves an individual coursework that will require for the students to demonstrate the practical skills they will have acquired with regards to the development of software applications using object-oriented programming, and an examination that will assess the students’ understanding related to analysis, design of system based on object-oriented principles and the more theoretical aspects of the module.

The individual coursework allows the students to demonstrate their ability to design and develop a software application using an OOP language, given a set a functional requirement. It will cover LO1, LO2, LO3 and LO4.

The in-class test allows the students to demonstrate their theoretical and practical knowledge and understanding of object-oriented concepts, object-oriented design and the basic principle of concurrency using an OOP language. It will cover LO1, LO2, LO3 and LO5.

**Assessment criteria**

In order to pass the module a student will have to demonstrate a detailed understanding of the design and development of applications based on Object Oriented Programming.

The criteria used to determine students’ performance will depend on the degree to which they are able to:

* Design a system using OOP for given a set of specification
* Develop GUI and console applications that can handle user input and output based on object-oriented development principles

To achieve higher marks a student should demonstrate:

* the ability of a very good usage of OOP principles
* appropriate design decisions for the classes and their hierarchy
* efficient coding and organisation
* usage of polymorphism, error handling, robustness, flexibility and maintainability and where appropriately the quality and justification for the chosen algorithms implemented or library used.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| ***Coursework*** | *50%* | *30%* |  | *Code and report* |
| ***In-class test*** | *50%* | *30%* |  | *Closed examination* |

**Synoptic assessment**

This module draws upon analytical, design, technical and development skills learned elsewhere in the course. This module therefore engenders synoptic learning and its assessment is inherently synoptic.

**Sources Essential reading**

Horstmann, C. (2016) *Big Java*, 6th edition. Wiley.

**Further reading**

Shildt , H. (2014) *Java. A Beginner’s Guide*, 6th Edition Eckel, B. (2006) *Thinking in Java*, 4th edition. Prentice Hall.

Bjarne, S. (2014), *Programming: Principles and Practice Using C++*, 2nd edition. Addison Wesley.

Weisfeld, M. (2013) *The Object Oriented Thought Process*, 4th edition. Addison-Wesley Professional

Patrick, HW, and Sundar, N. (2001) *On to Java*. Addison Wesley. Langr, J. (2005) *Agile Java*, Prentice Hall.

Robert, S and Wayne, K. (2011) *Algorithms*, 4th edition. Addison Wesley.

Gamma, E and Helm, R, and Johnson, R and Vlissides, J. (1994), *Design patterns: elements of reusable object-oriented software*. Addison Wesley.

King, KN. (2000) *Java Programming from the Beginning*. W.W. Norton & Company. [The Java Oracle site](http://java.sun.com/), <http://www.oracle.com/technetwork/java/index.html>

**Link to the online reading list**

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Robotic Principles | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College Design, Creative and Digital Industries; School Computer Science and Engineering | | |
| Module Leader(s): Nipuni Perera | | |
| Extension: | Email: | |
| Host course and course leader: BEng Smart Computer Systems | | |
| Status: Core - BEng Smart Computer Systems; Option – BSc Computer Science, BEng Software Engineering | | |
| Subject Board: COMENG | | |
| Pre-requisites: None | Co-requisites: None | |
| Study abroad: N/A | | |
| Special features: None | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an arrangement with an organisation(s) other than the University of Westminster: No | | |
| Summary of module content  This module introduces the fundamentals of robotics and focuses on selected topics pertaining to this discipline. Its introductory part overviews the nature of robotics and, related to it, challenges and issues. System modelling introduces techniques of deriving and computer implementation of models of dynamic systems with a special focus on kinematics of robots. Fundamentals of control cover the structure, basic analyses and real-time implementations of control systems. | | |

**Learning outcomes**

By the end of the module, the successful student will be able to:

LO1 Describe ethical, and health & safety issues related to practical use of robots; LO2 Use mathematical and software tools to model basic systems relevant to robotics; LO3 Analyse simple control feedback systems;

LO4 Implement and code simple control systems using a high-level development platform;

LO5 Solve forward and inverse kinematics problems with use of mathematical and software tools;

**Course outcomes the module contributes to**

BEng Smart Computer Systems L5.1, L5.2, L5.3, L5.7

**Indicative syllabus content**

The aim of this module is to teach the students the basics of robotics.

The introductory part of this module focuses on a general overview of the discipline, presenting latest trends, contemporary challenges and issues in robotics, and describing variety of specialised disciplines that relate to robots’ construction, exploitation, maintenance and relation with humans, including selected ethical, and health & safety issues. The general terminology related to robotics including robot classification, joints and degrees of freedom is introduced.

The main topics covered in this module focus on robot kinematics and on analysing control systems as relevant to robots. To this end the students learn underlying theory of robots and control systems. The teaching process is strongly supported by the use of appropriate software tools such as MATLAB, Python or Pygame to model the robots, and simulate their behaviour, as well as analyse control systems. The results of simulations are matched to and interpreted within the context of the underlying theory.

The kinematics of robots addresses both, forward and inverse kinematics problems in three- dimensional space. Students learn about popular tools used for solving these tasks. They are taught to combine formal analyses with simulations in order to predict the trajectory of the robot arm under specific movements of its individual joints, as well as defining trajectory of the arm and then imposing suitable actions on individual joints.

The introduction to control systems presents basics of control, the role of control systems, various configurations and their typical hardware requirements. While these topics are introduced in the context of robotics, they are generic enough to enable the students utilise their control engineering skills outside the robotics domain. The students learn the basics of measuring quality of control, stability of systems, and relating these concepts to frequency responses of the controlled systems. To help the students better understand the underlying formal analyses of control systems, teaching is supported by the use of MATLAB / Simulink simulations and deployment of software tools for analysing control systems.

**Teaching and learning methods**

The module runs each year in the second semester. Normally, there are four teaching hours per week. About half of the contact time are lectures. The remaining time are tutorials and practicals. Students are given unlimited access to software resources underpinning their study. Access to the hardware laboratory is restricted by the University policies including Health and Safety, and Security*.*

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 12 |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled | 8 |
| Supervised time in studio/workshop | Scheduled | 4 |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled |  |
| **Total Scheduled** |  | 48 |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | 200 |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

The nature of this module, which involves analytic and practical work, requires some diversification of the assessment methods. The students take three pieces of assessment. A practical assignment, in-class test and then the final exam.

The practical assignment requires the students undertaking a small-scale project, where they have to demonstrate the knowledge and ability to use appropriate software tools and techniques, up to the level that allows them completing engineering tasks that involve analysing robotic systems relevant to LO2, LO3 and LO4 learning outcomes. The closed book test focuses on the analytic site of the module (LO5 and parts of LO2) and on the non-technical contents of the module (LO1). The final exam revisits learning outcomes LO2 – LO5 and requires the students to demonstrate the depth of their learning and mastering the skills developed in this module.

**Assessment criteria**

To pass the module, the students must demonstrate basic competence in using the skills and methods specified in the learning outcomes of this module. To achieve higher marks, the students must demonstrate their ability to select the right tools to solve engineering problems relevant to this module. In addition, the students must demonstrate their ability to use the tools in previously unknown to them contexts.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| ***Assignment*** | *25* | *30* | *C* | *Technical report* |
| ***Test*** | *25* | *30* | *C* | *Closed exam* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Exam*** | *50%* | *30%* |  | *Closed exam* |

## Synoptic assessment

N/A

**Sources**

* László Keviczky, Ruth Bars, Jenő Hetthéssy, Csilla Bányász “Control Engineering”, ISBN 978-981-10-8296-2, Springer 2019
* László Keviczky, Ruth Bars, Jenő Hetthéssy, Csilla Bányász “Control

Engineering: MATLAB Exercises”, ISBN 978-981-10-8320-4, Springer 2019

* Niku S. B., Introduction to Robotics: Analysis, Control, Applications, 2nd Edition, John Wiley & Sons, 2011
* Nise N. S., Control Systems Engineering, 7th Edition, John Wiley & Sons, 2014
* Mark W. Spong, Seth Hutchinson, M. Vidyasagar, “Robot Modelling and Control”, John Wiley & Sons, 2006

**Link to the online reading list**

[https://rl.talis.com/3/westminster/lists/669B7922-1A15-0F69-406F-](https://rl.talis.com/3/westminster/lists/669B7922-1A15-0F69-406F-4D47F37DC268.html?draft=1&amp;lang=en-GB&amp;login=1) [4D47F37DC268.html?draft=1&lang=en-GB&login=1](https://rl.talis.com/3/westminster/lists/669B7922-1A15-0F69-406F-4D47F37DC268.html?draft=1&amp;lang=en-GB&amp;login=1)

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Sensors and Interfaces | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Sivaraman Ragu | | |
| Extension: | Email: | |
| Host course and course leader: BEng Smart Computer Systems | | |
| Status: Core - BEng Smart Computer Systems; Option - BSc Computer Science | | |
| Subject Board: COMENG | | |
| Pre-requisites: None | Co-requisites: None | |
| Study abroad: None | | |
| Special features: None | | |
| Access restrictions: Nobe | | |
| Are the module learning outcomes delivered, assessed or supported through an  arrangement with an organisation(s) other than the University of Westminster.  No | | |
| Summary of module content  The module provides an understanding of integrated smart sensing devices and emphasis on the integration of microcontroller-based systems with sensors. The students will learn about standard sensors and transducers and will be introduced to SMART sensing system concepts that have been developed for IoT applications and are built as IoT components that convert the real-world variable into a digital data stream for transmission. Apart from familiarising with smart communication protocols, the module enables the students to understand problems related to smart sensing systems and devices as well as experiment on standard interfacing circuits including microcontroller and serial bus interfacing. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Understand the working principles of various sensors and transducers and to develop a critical understanding of their applications;

LO2 Analyse the operational philosophy and specifications of smart sensors, their communications through bidirectional buses and interfaces, compatibility of smart sensors with digital systems based on MCUs;

LO3 Illustrate data transmission and communication procedures for effective processing, visualising and storing of measurement data, advances in MEMS technology, communication and interaction between smart sensors, communication protocols between intelligent sensors and how they differ from standard network protocols;

LO4 Demonstrate the operation of image sensing devices, acquisition and storage principles and image analysis, describe and apply the sequences of image analysis operations, representations, contrast techniques for extracting and representing features, edges, shapes, and textures;

LO5 Valuate appropriate serial interface configurations suitable for variety of data transmission requirements, taking into account the timing and protocol requirements of interfaces, behaviors of real-world signals and components.

**Course outcomes the module contributes to**

BEng Smart Computer Systems L5.1, L5.2, L5.3, L5.4, L5.5, L5.7

**Indicative syllabus content**

**Introduction to sensors:** Sensors for the measurement of different physical phenomenon such as temperature, light intensity, force, pressure acceleration; conversion of physical measurements to electrical signals; Applications of sensors: Sensors for communication, healthcare, transportation, industrial and aviation.

**SMART sensors and MEMS:** Introduction to smart sensors circuitry, smart sensors in IoT, application areas including communication networks, medical, control and instrumentation systems; Wireless Sensors and Actuator networks (WSAN) and its applications. Introduction to Microelectromechanical systems (MEMS); Integrated sensor design challenges and applications.

**Image Sensors and computer vision basics:** Image formation, Image sensing, pixel arrays, CCD and CMOS cameras; Mathematical operations for extracting image structures, morphological operators, filters, convolution, correlation; Edge detection operators, Gradient vector field, Laplacian operator and its zero-crossings; Image compression and storage

**Data Interfacing:** Introduction to standard interfacing circuits and Smart communication protocols, input/output device interfacing and component interfacing to microcontrollers, Serial interfacing, serial ports for data transmission, timing and protocol of common interfaces such as UART, SPI, USB, CANBUS, I2C, ZigBee, Bluetooth Applications of RS -232, USB interfacing, introduction to standard communication protocol for intelligent sensors, IEEE 1451.5.

**Teaching and learning methods**

The students will attend 2-hour lectures and 2 hours of tutorials/labs. The lectures will be interactive and are the principal means of introducing the relevant concepts and theory. The practical sessions are devised to reinforce these concepts through hands-on problem-solving

sessions. The tutorial sessions are arranged to help the students with the practical aspects as well as theoretical understanding, enabling them to achieve the required learning outcomes. The students will be encouraged to access blackboard for lab work assistance, practise questions and solutions as well as relevant announcements regarding lectures/tutorials and assessments.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 28 |
| Seminar | Scheduled | 8 |
| Tutorial | Scheduled |  |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled | 8 |
| Supervised time in studio/workshop | Scheduled | 4 |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled |  |
| **Total Scheduled** |  | **48** |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | **200** |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

Tests taken under exam conditions assess the students’ understanding and knowledge of the syllabus as they go through the module. Scripts are returned with marks and feedback, and answers are discussed in class to assist learning.

Assessment of the laboratories measures the students’ ability to work singly or in small groups to apply the theory in practice and draw correct conclusions from their observations.

The final test checks the students’ understanding of the underlying theory covered in the module and their ability to apply it to a selection of problems. The students are also expected to have awareness of appropriate codes of practice, regulatory practice, industry standards and protocols.

Formative assessment will be implemented by giving the students practice tests, laboratory sessions and documented demonstrations for which the marking will not contribute to the module marks but will provide feedback on the students’ strengths and weaknesses in preparation for the later summative assessments.

**Assessment criteria**

To pass the module, the student must demonstrate basic competence in using the skills and methods taught. To achieve higher marks, the student must be able to select and apply these skills more widely and in different ways.

The extent to which the student has demonstrated achievement in LOs 1, 2, 3 and 5 are evaluated through written test assessments including formative tests.

The extent to which the student has demonstrated achievement of LO 4 and part of LO5 are evaluated in the laboratory through demonstrations of their practical work and their completed laboratory handouts which document their designs, results and conclusions from their observations.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| ***In class test*** | *25* | *30* | *C* | *In-Class test* |
| ***Laboratory*** | *25* | *30* | *C* | *Laboratory Practical* |
| ***Exam*** | *50* | *30* |  | *Closed book exam* |

**Synoptic assessment**

This module draws on technical, analytical and developmental skills learned elsewhere in the course. This module therefore engenders synoptic learning and its assessment in inherently synoptic.

**Sources**

*Stanley M, Lee J: Sensor analysis for the Internet of things, Morgan & Claypool 2018, ISBN: 1-68173-288-2*

*Nihtianov S, Luque A: Smart sensors and MEMS: intelligent devices and microsystems for industrial applications, Second edition, Woodhead Publishing, 2018, ISBN : 0-08-102055-4*

*Webster, J.J, Eren, H, Measurement, Instrumentation and Sensors Handbook, CRC Press Handbook, 2014, ISBN: 9781439848838.*

*Gonzalez RC, Woods RE. Digital Image Processing. 3rd Edition. Pearson International, 2008, ISBN : 9780131687288.*

*Soille P. Morphological Image Analysis. 2nd Edition, Springer-Verlag, 2003, ISBN: 3-540- 42988-3.*

*Bates, M, Interfacing PIC microcontrollers: embedded design by interactive simulation, Newnes, 2006, ISBN: 978-0-7506-8028-8*

## Further Reading:

*IEEE Std 1451.4-2004: IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*

[*Kirianaki*](https://www.google.co.uk/search?tbo=p&amp;tbm=bks&amp;q=inauthor%3A%22Nikolay%2BV.%2BKirianaki%22)*, N.V,* [*Yurish*](https://www.google.co.uk/search?tbo=p&amp;tbm=bks&amp;q=inauthor%3A%22Sergey%2BY.%2BYurish%22)*, S.Y,* [*Shpak*](https://www.google.co.uk/search?tbo=p&amp;tbm=bks&amp;q=inauthor%3A%22Nestor%2BO.%2BShpak%22)*, N.O and* [*Deynega*](https://www.google.co.uk/search?tbo=p&amp;tbm=bks&amp;q=inauthor%3A%22Vadim%2BP.%2BDeynega%22)*, V.P, Data Acquisition and Signal Processing for Smart Sensors, Wiley, 2002.*

**Link to the online reading list**

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Server-side Web Development | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value:10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering School of Computer Science and Engineering | | |
| Module Leader(s): Chathura Sooriya-Arachchi | | |
| Extension: | Email: | |
| Host course and course leader: BSc Computer Science | | |
| Status: Option - BSc Computer Science; BEng Software Engineering; BSc Business Information Systems | | |
| Subject Board: COMENG | | |
| Pre-requisites: None | Co-requisites: None | |
| Study abroad: An alternative assessment can be produced to evaluate whether Study Abroad Students have passed the Learning Outcomes. | | |
| Special features: None | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an  arrangement with an organisation(s) other than the University of Westminster. Yes | | |
| Summary of module content  This module covers the design and implementation of commercial dynamic web applications from a server-side programming and database perspective. It is suitable for students with a strong interest in SQL, web programming, HTML, CSS and browser scripting. A server-side language is covered to the depth required for implementing high-quality fully functional web- enabled database applications that fittingly support an organisation’s business processes. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Identify and argue merits of alternative approaches and methods for creating dynamic web sites for e-business;

LO2 Critically evaluate various Web-related technologies and select the appropriate ones to design and develop a functional database-driven Web application to support the business processes of an organisation;

LO3 Demonstrate a detailed understanding of the design and implementation of a server- side processing solutions involving database access (Create, Read, Update and Delete), form handling, data persistence, user authentication and session handling;

LO4 Design and develop a user interface for the database-driven Web application using appropriate technologies;

LO5 Critically evaluate issues related to emerging web application technologies and content management systems.

**Course outcomes the module contributes to**

N/A (the module is not core in any of the courses offered)

**Indicative syllabus content**

* Introduction and thorough discussion of various approaches to the development of dynamic web applications for e-commerce and e-business e.g. agile methods.
* Introduction and thorough discussions of several techniques to support the design of dynamic database-driven web applications e.g. Entity-Relationship modelling, storyboarding, wireframing, prototyping.
* Introduction and thorough discussion of various Web-related technologies.
* Separation of concerns e.g. front-end/back-end; logical/physical, three-tier/multi-tier architectures, client-side/server-side.
* Build on previously acquired knowledge on cascading style sheets – style, look and feel, elements and positioning.
* Development of a back-end database to make it accessible from the Web.
* Employment of server-side scripting technology with embedded SQL to connect to database and provide dynamic content.
* Server-side programming constructs and techniques e.g. array of records, GET and POST methods, session arrays, handling forms, database transactions and error handlings
* Implementation of the 4 database functions (create, read, update and delete**)** to and from a web application.
* Introduction to iterative testing within an agile development framework.
* Introduction of alternative approaches involving the setting up and customisation of content management systems; compare and contrast different content management systems.
* Exposure to data analytics to improve performance of e-commerce applications through heightened customisation.

**Teaching and learning methods**

This module will follow a project-driven approach and will be taught by a combination of

lectures, tutorial sessions and self-study activities.

In the lectures (2 hour weekly), you will be introduced to underlying concepts, theories and

principles which will provide the foundation for the tutorial activities.

The tutorials (2 hours weekly) will be central to the learning, providing the opportunity to work on the practical development of a web-based application. Further, the tutorials will offer the opportunity to interact with your fellow students and tutors to gain understanding of the topics and confidence in communicating this.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 24 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 24 |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled |  |
| Practical Classes and workshops | Scheduled |  |
| Supervised time in studio/workshop | Scheduled |  |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled |  |
| **Total Scheduled** |  | **48** |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | **200** |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

The summative assessment strategy involves two development-driven phase tests which will assess the students understanding of the web development constructs and the technical skills gained in the module and their ability to develop and implement web applications capable of connecting to a database.

The first phase test allows students to demonstrate their understanding of the key technical design and implementation issues of a simple web server solution. It allows students to demonstrate their abilities to justify the appropriate technologies suited to design and develop a simple dynamic web application with an appropriate interface as well as to engage with the implementation of web-database connectivity (create and read), and user authentication (LO3 and LO4).

The second phase test allows students to demonstrate their understanding of the key technical design and implementation issues of a sophisticated web server solution. It allows students to demonstrate their abilities to justify the appropriate techniques and methods suited to develop a complex dynamic web application as well as to engage with the implementation of web-database connectivity (create, read, delete and update), advanced browser-server interaction, data persistence and session handling. It also allows them to

show the extent of their understanding of the issues related to emerging web application

technologies (LO1, LO2 and LO5).

Formative assessments will be embedded within the teaching and learning activities of the module. At every step of the development process, students will be required to regularly demonstrate the extent of the development of their prototype to the rest of the class so that to get feedback from their peers but also from the members of the module team. This will result in constructive feedback which they can then re-invest in their development. In addition, a formative test session can be conducted to enable the students to answer questions on their thinking behind the developmental aspects of the module and be given feedback before the summative tests.

**Assessment criteria**

In order to pass the module, students must demonstrate a detailed understanding of web

server-side techniques.

The criteria used to determine students’ performance will depend on the degree they are able to:

* Select and apply sound methods and techniques in the design and implementation of both simple and sophisticated server-side applications;
* implement and explain appropriate server-side functionality in dynamic Web applications;
* implement and explain ways to connect web-based systems to server-side databases and to create, retrieve, delete and update records into and from the database;
* present sound and complete arguments on the merits of alternative technologies used for creating dynamic web applications;

Meeting the above criteria will mean a pass mark. Better students are expected to be able to provide fuller answers; they should be able to demonstrate an insight into the topics covered and to produce innovative answers to more challenging issues.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| ***In-class Test 1*** | *50* | *30* |  | *Written exam – coding exercise* |
| ***In-class Test 2*** | *50* | *30* |  | *Written exam- coding exercise* |

**Synoptic assessment**

This module draws on analytical, design and development skills learned elsewhere in the course. This module therefore engenders synoptic learning and its assessment is inherently synoptic.

**Sources**

*Adrian W West & Steve Prettyman. Practical PHP 7, MySQL 8, and MariaDB Website Databases A Simplified Approach to Developing Database-Driven Websites. Second edition. Apress. 2018.*

*Andrew Caya. Mastering the faster web with PHP, MySQL and JavaScript: develop state-of- the-art web applications using the latest web technologies. Packt Publishing. 2018.*

*Robin Nixon. Learning PHP, MySQL, & JavaScript: with jQuery, CSS & HTML5. O'Reilly. 2018.*

*Porter Scobey & Pawan Lingras. Web programming and Internet technologies: an E- commerce approach. Second edition. Jones & Bartlett Learning. 2018.*

**Link to the online reading list**

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: Software Development Group Project | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): John Sriskandharaja / Banuka Athuraliya | | |
| Extension: | Email: | |
| Host course and course leader: BSc Computer Science | | |
| Status: Core – BSc Computer Science, BEng Software Engineering | | |
| Subject Board: COMENG | | |
| Pre-requisites: | Co-requisites: | |
| Study abroad: Yes | | |
| Special features: Industry based project | | |
| Access restrictions: Database and Programming experience | | |
| Are the module learning outcomes delivered, assessed or supported through an arrangement with an organisation(s) other than the University of Westminster: Yes | | |
| Summary of module content:  In this module students gain practical work experience through their participation in a team that develops a prototype for a real-life software application suggested by industry contacts. The module takes the students through the different managerial and technical steps of software development. Students gain experience in managing a team and deliver software iteratively by using an agile approach and receiving feedback from industry contacts. The module also provides students with experience in team communication and ways to overcome any problems, as well as the opportunity to reflect on professional issues such as quality of project documentation, cybersecurity, ethics and code of conduct. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Work effectively with an external client through the management and development of a real-life project using modern agile software lifecycle approaches;

LO2 Work as a member of a software development project team demonstrating collegiality and professional standards of conduct in communication, time management, project documentation and version control;

LO3 Identify, use and evaluate relevant frameworks and technologies when developing software including the application of appropriate testing methods and tools to evaluate software performance;

LO4 Consider the ethical and legal relevant issues of a software project, including issues of intellectual property and data protection;

LO5 evaluate security risks and demonstrate how computer security decisions affect software development;

**Course outcomes the module contributes to**

BSc Computer Science L5.1, L5.2, L5.5, L5.6, L5.7, L5.8, L5.9

BEng Software Engineering L5.1, L5.3, L5.5, L5.8, L5.7, L5.9

**Indicative syllabus content Software project development**

* + Models of software projects, estimating the effort;
  + Software development lifecycle;
  + Requirements analysis, user stories;
  + Modelling a system: the different needs of the data, the human computer interface and what lies in between;
  + Introduction to agile approach.

**Professional behaviour**

* + The needs of others; boundaries of rights and responsibilities;
  + Legal issues as they apply to software development, data protection (GDPR), intellectual property;
  + Ethics: the project team, the wider world, and life-long development; privacy; ethical clearance and ethics unclear: dilemmas;
  + Professional behaviour in a global, distributed, environment.

**Quality Assurance**

* + Safety, reliability, availability, resilience and security;
  + Testing and version control; tools; evaluation beyond software testing: critical evaluation of sources, research, documentation, communication;
  + Security requirements: adding a security component versus adding security in all components;
  + Threats and counter-measures: external security threats, internal security threats and security threats from user ignorance.

**Teaching and learning methods**

This module is delivered in a mixture of presentations, lectures (2 hours) and tutorials (2 hours). The module is supported by industry contacts who introduce the client project to the students, deliver talks on industry practices such as integration and deployment, and provide feedback on the students’ work during a scheduled event in the teaching calendar and the

final presentation of the best students’ work. The development of the project is student driven with weekly formative feedback from the tutors during the tutorials on the projects’ progress and the team-centred stand-ups. The taught material delivered during lectures focuses on managing development projects using agile methods, system modelling, HCI considerations and discussions on professional, legal and ethical issues. Part of the taught material is formative assessed during the tutorials through short on-line tests followed by feedback from tutors. Outside the formal class time, there is online support via Blackboard.

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 22 |
| Practical Classes and workshops | Scheduled | 26 |
| **Total Scheduled** |  | **48** |
|  |  |  |
| Module and coursework -based general study | Independent | 82 |
| Preparation for Assessment | Independent | 70 |
| **Total Independent Study** |  | **152** |
|  |  |  |
| **Total student learning and teaching hours** |  | **200** |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

Being a practical module by its nature, the module encompasses a range of applied knowledge and practical skills that students should have gained from other core modules of their courses. Thus, the module has a balanced mixture of formative summative, and synoptic assessment.

Formative assessment: to reinforce learning, formative assessment is provided on a weekly basis. There are two types of formative assessments: a) A set of formative on-line tests on topics covered during the lectures. Solution are provided to get feedback from a group supervisor; b) Feedback on the work presented weekly by the group to the group supervisor. At the end of the module the best pieces of work are selected for presentation to the industry contacts setting the brief, with a non-credit bearing award to the best implementation as selected by our industry contacts.

Summative Assessment: The module consists of two summative assessments: CW1 “Design” and CW2 “Implementation, Evaluation and Testing”. In both pieces of coursework students receive group and individual marks, based on the individual contribution of a student to a group work which is also monitored by a tutor (preparation for assessment 30 hours).

CW1 comprises analytic and synthetic work on analysing and presenting data relevant to the project, report, software documentation. The students work as a team, and all contribute to the specified group elements of the coursework. Students need to indicate that each contributed equally to each of the elements. CW1 also has individually assessed elements and includes a collection of the students’ weekly stand-up, reflection on the feedback received and feedback given to their team members.

CW2 comprises practical work on the implementation of the design, its evaluation and testing, report, software documentation, presentation, demonstration. Students receive individual mark for the implementation of their project part, as well as their weekly stand-up and feedback given. Students receive group mark for the overall quality of the software product developed

and their contribution in ensuring a smooth integration of the various parts (preparation for assessment 40 hours)

Note that the module is designed in such a way that consistent and continuous work during the scheduled tutorial and timely management of group project, including all required elements of the project documentation, will contribute a significant proportion of the final deliverables, allowing the balanced spread of students’ time and efforts and thus releasing stress from the final stages prior to the submission.

Demonstration of the work to the group supervisor is an essential and part-fulfilment of the assessment.

**Assessment criteria**

Two courseworks are group practical work, with the first aiming at the specification and design of the software (LO1, LO2, LO3, LO4) and the second allowing for the integration of design and development skills (LO1, LO2, LO3, LO4, LO5). Both courseworks will be done in groups with individual work per student. In CW1 higher marks will be achieved by demonstrating higher level of analytical skills in consistent design solution that meets the requirements with strong evidence-based and research informed demonstrations of professionalism. In CW2 higher marks will be achieved by the full and consistent implementation of the design reflecting the functionality of the system, advanced implementation of classes and their relationships, implementation and use of the database, use of advanced programming techniques, implementation of the graphical user interface to meet the principles of HCI and usability, and the completeness of evaluation and testing. Additionally, students will gain higher marks in both courseworks for effective team-work.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| ***Group Coursework 1: Project Design*** | *40* | *30* |  | *Software Project: report, presentation, project documentation* |
| ***Group Coursework 2:***  ***Project Implementation evaluation and Testing*** | *60* | *30* |  | *Practical work on Software Project: report, presentation, demonstration, project documentation* |

**Synoptic assessment**

This module draws on technical and analytical skills learned elsewhere in the course. This module therefore engenders synoptic learning and its assessment is inherently synoptic.

**Sources Essential reading**

Ian Sommerville (2015) [Software Engineering (10th Edition)](https://www.amazon.com/dp/0133943038/ref%3Drdr_ext_tmb), Pearson

Rumpe B. (2017**)** Agile Modelling with UML: Code Generation, Testing, Refactoring, Springer

R. C. Martin (2008), Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall

British Computer Society Code of Conduct (www.bcs.org)

**Further reading**

R. C. Martin (2005), Agile Estimating and Planning, Prentice Hall, Prentice Hall

Visser, J. (2016) Building Maintainable Software: Ten Guidelines for Future-Proof Code (O'Reilly Media, Inc; Java edition)

Bainbridge, D (2009), Introduction to IT Law, Pearson Barger R (2008) Computer Ethics, CUP

**Link to the online reading list**

|  |  |  |
| --- | --- | --- |
| MODULE PROFORMA | | |
| Full module title: XR Multimodal Interaction | | |
| Module code: | Credit level: 5 | Length: 1 Semester |
| UK credit value: 20 | ECTS value: 10 | |
| College and School: College of Design, Creative and Digital Industries; School of Computer Science and Engineering | | |
| Module Leader(s): Pumudu Fernando | | |
| Extension: | Email: | |
| Host course and course leader: BSc Computer Science | | |
| Status: Option - BSc Computer Science, BSc Computer Games Development; Westminster Plus Elective | | |
| Subject Board: COMENG | | |
| Pre-requisites none | Co-requisites: none | |
| Study abroad: No alternative CW required | | |
| Special features: IIT Franchise Module | | |
| Access restrictions: None | | |
| Are the module learning outcomes delivered, assessed or supported through an  arrangement with an organisation(s) other than the University of Westminster: No | | |
| Summary of module content  This module introduces students to concepts of XR (Virtual, Augmented, and Mixed Reality) interaction, and uses an industry standard games engine to develop assets, properties, controllers and scripting for creating interactive rich media content. This is part of the “Usability and Interaction” and “Games and Computer Graphics Development“ themes for Computer Science, but is open to all courses with no pre-requisite. Supported coursework path is the production of a VR interactive media product. | | |

**Learning outcomes**

By the end of the module the successful student will be able to:

LO1 Demonstrate an applied understanding of the XR hardware and software ecosystem, XR UI concepts and the underlying principles of 3D UI and apply appropriate techniques and tools to create XR interactive media;

LO2 Incorporate 3D models, animation, and graphical content in an industry standard games platform/engine and also SDK for XR development;

LO3 Demonstrate an applied intermediate understanding of XR sensor concepts, properties,

controllers and use of scripting techniques to create XR Interactions;

LO4 Demonstrate an applied understanding of scripting techniques to incorporate interactive navigation behaviours for XR concepts and SDKs;

LO5 Work individually to develop an XR interactive media product;

LO6 Communicate design concepts by oral and visual means and provide documentation for an XR interactive media product in written form.

**Course outcomes the module contributes to**

N/A (the module is not core in any of the courses offered)

**Indicative syllabus content**

* XR Ecosystem, hardware, software, platforms, frameworks, SDKs, toolsets, tookkits APIs, controllers, support, history.
* XR/VR User Interfaces (UIs) and interaction styles, introduce applications of XR UIs, gamified interfaces, mobile and web applications, etc. Discuss principles of designing Usable XR UIs. Understand the design space.
* XR(VR) and Stereo Visual Cues, basic concepts of VR representation.
* 3D world representation and 3D space transformation, expansion of 3D Interactive Media Development.
* Key concepts of 3D Asset Sourcing and Manipulation for XR, expansion of Game

Design and Asset Creation, with specific use of 3D assets for XR(VR).

* Understanding XR Visual effects and lighting, shading (composing and colour grading), texturing mapping (UV texture layout, coordinates, mapping) and paint effects. Brief introduction to lighting (illumination model, light types and properties) and shading (mental ray shading). Expansion of Game Design and Asset Creation for XR(VR).
* XR(VR) animation, inverse kinematics for XR/VR
* Basic XR(VR) Coding, control 3D geometry and animation properties with the use

of appropriate scripting language.

* Basic XR(VR) Navigation, control a 3D camera system and enable various POV systems and interactions. Combine navigation with animation as required.
* Mobile/Networked/Social VR
* Current VR UI Research, investigation of latest concepts and work, especially

from <http://ieeevr.org/>

* Current XR/VR UI platforms and delivery, build to platforms and begin to take XR systems and hardware/software into account, leading to XR Multimodal Interaction module.

**Teaching and learning methods**

This module is taught using a combination of lectures or workshops and laboratory-based tutorial sessions or workshops. The lectures or workshops include demonstration of the syllabus and the students are given a framework based on which they can classify and evaluate the course materials in their subject. During tutorials or workshops, students are introduced to industry-standard tools that are used for the development of a 3D scene and 3D animated and interactive content, offering experience of applying knowledge acquired during lectures through practical exercises and case studies. Students are provided with a set of practical exercises to be completed both during and outside the scheduled tutorial times. Immediate feedback on proposed solutions to the exercises is provided in the tutorials with discussion of problems providing formative feedback on students’ progress and understanding.

Students will have the opportunity to present their design concept and their proposed approach based on which they will be provided with formative feedback and update their design. Students will work individually, to consolidate the concepts covered and enhance their practical skills in research, writing, speaking and presenting.

There are 152 hours allocated to Independent study, this forms a very important part of the module where as a student you are expected to use this time to prepare and develop your skills and maximise your time by using the practical classes as a point of contact to clarify any issues you may have, thus maximising your learning. Typically, as a guide you should break down your time for independent study in the following way:

Webinars: Guided Independent Study – 12 hours Preparation of Assessment – 12 hours per CW Preparation of Peer Review – 4 hours per CW Development work – 48 hours per CW

Lab based development work – 12 hours per CW

|  |  |  |
| --- | --- | --- |
| **Activity type** | Category | Student learning and teaching hours\* |
| Lecture | Scheduled | 12 |
| Seminar | Scheduled |  |
| Tutorial | Scheduled | 12 |
| Project supervisor | Scheduled |  |
| Demonstration | Scheduled | 6 |
| Practical Classes and workshops | Scheduled | 6 |
| Supervised time in studio/workshop | Scheduled | 6 |
| Fieldwork | Scheduled |  |
| External visits | Scheduled |  |
| Work-based learning | Scheduled | 6 |
| **Total Scheduled** |  | **48** |
|  |  |  |
| Placement | Placement |  |
| Independent study | Independent | 152 |
| **Total student learning and teaching hours** |  | **200** |

\*the hours per activity type are indicative and subject to change.

**Assessment rationale**

The coursework rationale is to introduce the students to the practical nature of XR Interfaces, sensors, and development. It realizes the above by setting two pieces of coursework that assess student knowledge and understanding in different aspects of composing an XR interactive media product:

* Coursework 1 - assesses the student skills in working to plan and visualise a XR interactive media concept that addresses specific technical requirements, translate these into a practical sequence and compose the required XR scene and content (the content will be later animated and interactive by adding appropriate behaviours to it) using industry standard tools and products. In addition, it assesses the student’s skills presenting and communicating their design concept efficiently. (Learning outcomes 1, 2, 5, 6)
* Coursework 2 - assesses the student practical ability to use appropriate scripting language to create required animation and interactivity to a proposed XR interactive media product. In addition, it assesses the student’s skills documenting their implementation in a report using a professional style and their entrepreneurial skills by the creation of a working video demo of their project. (Learning outcomes 3-6).

Both pieces of coursework are assessed by means of extended tutorial exercises, to be completed both during and outside tutorials. Milestones are set and coursework components are submitted progressively and discussed during tutorials. There are numerous of benefits following this process: it allows formative feedback to be provided; it allows reflection and produces progressive iterations of the design; it aids continuous engagement; it makes it possible to track engagement, and peer review is done to develop warranting and connoisseurship of artefacts. Summative assessment is applied on the final iteration of the XR interactive media product.

**Assessment criteria**

The student will demonstrate achievement of the learning outcomes by being able to:

* + Investigate, collect, and synthesize XR software, hardware, and concept information

to present a coherent strategy for an XR product.

* + Compose an XR interactive media product following given technical requirements without restricting creativity.
  + Using appropriate tools and scripting language to create XR animated, interactive media content, smoothly integrate it in a XR interactive media product and deploy it for multiple platforms and devices.
  + Communicate design concepts by oral and visual means.
  + Write documentation of a XR interactive media product.

To achieve higher marks than the threshold, students must demonstrate a greater degree of expertise and knowledge, and in particular to demonstrate knowledge of economic

programming skills, creative technology skills, material beyond that covered in class, and self-evaluation.

**Assessment methods and weightings**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment name** | **Weighting**  **%** | **Qualifying mark %** | **Qualifying set** | **Assessment type (e.g. essay, presentation, open exam or closed exam)** |
| ***Coursework 1***  ***– XR Investigation and prototype*** | *40* | *30* |  | *Coursework presentation* |
| ***coursework 2***  ***– XR Interactive Media Product*** | *60* | *30* |  | *Coursework report* |

**Synoptic assessment**

This module draws on technical, analytical, research and presentation skills learned elsewhere in the course. This module therefore engenders synoptic learning and its assessment is inherently synoptic.

**Sources**

**Link to the online reading list**

[https://rl.talis.com/3/westminster/lists/2FCEE074-69F5-5CB2-B4CF-2074B2602D26.html?lang=en-](https://rl.talis.com/3/westminster/lists/2FCEE074-69F5-5CB2-B4CF-2074B2602D26.html?lang=en-GB) [GB](https://rl.talis.com/3/westminster/lists/2FCEE074-69F5-5CB2-B4CF-2074B2602D26.html?lang=en-GB)

[https://rl.talis.com/3/westminster/lists/FFA649D0-172E-2F19-C52B-EFD7D3DB5C18.html?lang=en-](https://rl.talis.com/3/westminster/lists/FFA649D0-172E-2F19-C52B-EFD7D3DB5C18.html?lang=en-GB&amp;login=1) [GB&login=1](https://rl.talis.com/3/westminster/lists/FFA649D0-172E-2F19-C52B-EFD7D3DB5C18.html?lang=en-GB&amp;login=1)