Module Specification

Logic to Artefact

Version: 2021-22, v1.0, 09 Mar 2021

Contents

Part 1: Information

Module title: Logic to Artefact

Module code: UBLLW1-15-M

Level: Level 7

For implementation from: 2021-22

UWE credit rating: 15

ECTS credit rating: 7.5

Faculty: Faculty of Environment & Technology

Department: FET Dept of Architecture & Built Environ

Partner institutions: None

Delivery locations: Frenchay Campus

Field: Architecture and the Built Environment

Module type: Project

Pre-requisites: None

Excluded combinations: None

Co-requisites: None

Continuing professional development: No

Professional, statutory or regulatory body requirements: None

Part 2: Description

Overview: The module is designed for complete novices in computational design platforms as one of 2 Bootcamp-based modules. Logic to Artefact is both an introduction and revision of computational design fundamentals. The Bootcamp fast-paced module is to prepare for the building systems scale design exploration. For the Spring cohort intake, this module must be the introductory module , on which everything is built. The module focuses on computation for fabrication to help

students use automated fabrication techniques and develop intelligent systems at the assembly detail scale. The module is designed to be a standalone module for MArch elective courses and CPD students.

The Bootcamp fast-paced module is to prepare for the building and system scale design exploration. A series of fast-paced lab-based tutorials introducing computational methods and cutting-edge computational tools ramp up the cohort's knowledge foundation. Tracking these technical learnings' understanding will be evaluated individually as a response to a primer brief.

In tandem, the students will be introduced to literature underpinning form-finding and system craft as part of the History and Theory thread woven throughout the programme. The students become familiar with how historical form-finding analogue methods and pattern design language has been systematically interpreted in computation design.
The tuition targets computational microscale design methods, such as interior architecture, building systems, and architectural scale interventions. Microscale design abstractions generally focus on material testing, system validation, and robotic fabrication methodologies inspired by natural and biological systems. Rapid prototyping abstractions would target modelling large-scale digital craft using cutting edge robotic and automated fabrication, and smart material design.

These technical skills and theoretical understandings begin in this module and are explored more deeply elsewhere in the programme.

Features: The fast-paced learning in this module has two tandem tracks 1) individual technical learning and 2) team work focussed on theory. The individual authored contribution considers each students' understandings of the fast-paced tuition of computational methods and the ability to tap into the open-source community to find solutions agilely. The required teamwork submission is evaluated as a single output to appraise the students' interpretation of the field's theoretical relationship. The group work is an icebreaker for the cohort, enabling them to collectively climb the learning curve and introduce them to this interdisciplinary field's collaborative nature.

Educational aims: Understand the different types of system form-finding, modelling, testing and validation with computational tools.
Investigate and Synthesise theories underpinning digital pattern language and form-finding.
Develop skills and knowledge of computational fundamentals of visual programming, system validation and expand their knowledge by tapping into FET's resources and the online open-source community
Investigate form-finding, material testing methods and fabrication experimentation with large-scale automated fabrication machines.
Develop skills in the visual and verbal representation, artefact curation, and simulation of building scale problems and solutions.
Works creatively and effectively within a team, supports or is proactive in leadership, negotiates in a professional context and manages conflict and creative differences. Proactively seeks to resolve conflict.

Outline syllabus: The module runs as a short, intense skills lab-based module with a theoretical thread addressing pattern language and form-finding. As a 4-week 15-credit module, teaching will be delivered two days a week.

As an introductory course, there will be a general lecture on the theoretical and historical overview of the topic, focusing on translating logic and algorithms into a physical artefact in academia and practice. Closer to the end of the term, one-to-one clinical hours will be scheduled to ensure students are supported as they climb the learning curve.

This module's simulation strand is delivered in a series of intense day-long and half day-long sessions in a computer lab. This learning's Bootcamp nature aims to ramp up and bring up the cohort's programming knowledge base to fundamental skills. The programme used in this module is defined by the tutors teaching the module, and the current practices. The programming topics covered are:

Fundamentals of visual programming
Form finding and Generative Design
Simulation platforms (structural, environmental, acoustical, etc.)

Designing for Mixed reality process (Virtual Reality, Augmented reality, etc.)

The fabrication strand is delivered similarly to the simulation strand, namely, Bootcamp delivery. The tuition focuses on current modes of dissecting a system assembly for manufacturing, using automated machines to create a designed artefact, within a microscale context.

The historical, theoretical and critical understanding of pattern language and form-finding will be delivered as lectures and seminars over two weeks. The lectures and readings focus on the wealth of literature and methodologies dating back to the 1960s. The students are invited to read and distil seminal texts to be discussed in the seminar sessions. In a small group of 2 to 3, the team is invited to translate an algorithm in an analogue (non-computational) exercise.

Part 3: Teaching and learning methods

Teaching and learning methods: The fast-paced learning in this module is delivered on two tracks:

1. Lab-based practical skills:

The practical lectures, exercises and primer project are designed to facilitate competency acquisition through applied and indirect learning, building knowledge by introducing the new subjects and reinvestment of gained skills of visual programming and digital fabrication.

2. Theoretical Seminar and self-directed study:

This track enables students to support their independent learning by exploring more profound issues of computation design and receiving formative feedback. In these seminars, students are exposed to the long track of computational design accumulated during the past two decades and are encouraged to build on that body of work. The introduction of manual logic and form-finding exercises and the invitation to abstract the process into visual representation allows students with non-design and design background to understand and communicate complex information.

For both aspects, studies conducted outside of contact hours is essential to complete the assigned work successfully. Students will be expected to come prepared for the module sessions with work-in-process and seminal readings. Feedback will be in the form of direct verbal and/or written.

Marking criteria and assessment format will be indicated on the project brief will be accessible to the students at the beginning of each project. Students' work will also be exposed to critical peer evaluation through discussion.
Presentations by the students will enable peer learning and help students develop the skills and capabilities to analyse problems, negotiate, make decisions, and present solutions to problems collaboratively.

Module Learning outcomes:

MO1 Interpret Pattern Language theory and form-finding methods and recognise the interrelationship with computation design processes' in academia and practice.

MO2 Identify and apply the types and fundamentals of visual programming, material testing, and digital fabrication methods to represent microscale design, in response to a primer brief.

MO3 Apply researched knowledge of pattern and form-finding algorithms in design processes to produce a well-crafted artefact (simulation/fabrication).

MO4 Flexibly and creatively select appropriate computational and fabrication methods by proactively undertaking substantial investigations to address an architectural system design.

MO5 Present an interpretation of fundamental Pattern Language and form-finding theories in a crafted artefact, graphical and verbal illustration at a high level of abstraction, arguing from competing perspectives.

Hours to be allocated: 150

Contact hours:

Independent study/self-guided study = 120 hours

Studio sessions = 30 hours

Total = 150

Reading list: The reading list for this module can be accessed at readinglists.uwe.ac.uk via the following link https://uwe.rl.talis.com/modules/ubllw1-15-m.html

Part 4: Assessment

Assessment strategy: The assessment strategy adopted by this module involves a mix of practical skills assessment, and a verbal and graphical presentation to reflect on key complexity theories applied in design.
Individual assessment is a practical portfolio designed to evaluate students' fundamental practical coding skills and apply them to generate solutions. The portfolio documentation will demonstrate the student's ability to define inputs and select an algorithm towards finding a solution as an output, for the individual practical design project.
For the team work project the students are expected to design and present a group visual presentation (poster or PowerPoint) responding to the theoretical strand brief. The students must appreciate the criticality of communicating complex data and logic visually.

Resit Strategy: The portfolio will include the revised individual work to the same brief, to evaluate students' fundamental practical coding skills and application. It will also include an individual experiment with a reduced scope to acknowledge the change form group work to an individual resit. This revised experiment will include the theoretical component as part of the design process. The student will also be asked to reflect on what has been lost, gained, and learned by working individually or in a team.

Assessment criteria will be made available to the students, along with each assignment brief.
Feedback: there will be peer and tutor feedback throughout the module critiques.

The students will be invited to provide self-assessment. Written feedback on completion of the projects.

Assessment components:

Final Project - Component A (First Sit)
Description: Individual Practical Project (30 pages)
Weighting: 50 %
Final assessment: No
Group work: No
Learning outcomes tested: MO2, MO3, MO4

Presentation - Component A (First Sit)
Description: Theoretical Teamwork Project (10-15 min presentation)
Weighting: 50 %
Final assessment: Yes
Group work: Yes
Learning outcomes tested: MO1, MO4, MO5

Portfolio - Component A (Resit)
Description: Portfolio (40 pages): requires and updated failed component and merging independent practical work and theoretical investigation, the portfolio should include a reflection of loses, gains, and learnings from individual and teamwork, and presented coherently.
Weighting: 100 %
Final assessment: Yes
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4, MO5

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Computational Architecture [Sep][FT][Frenchay][1yr] MSc 2021-22

Computational Architecture [Sep][PT][Frenchay][2yrs] MSc 2021-22

Computational Architecture [Sep][FT][Frenchay][1yr] MSc 2021-22

Architecture [Sep][FT][Frenchay][2yrs] MArch 2021-22

Computational Architecture [Sep][PT][Frenchay][2yrs] MSc 2021-22

Architecture {Apprenticeship-UWE} [Sep][FT][Frenchay][3yrs] MArch 2019-20

Architecture [Sep][PT][Frenchay][3yrs] MArch 2019-20