

BSc(Hons) Software Engineering for Business

Year 1 Modules

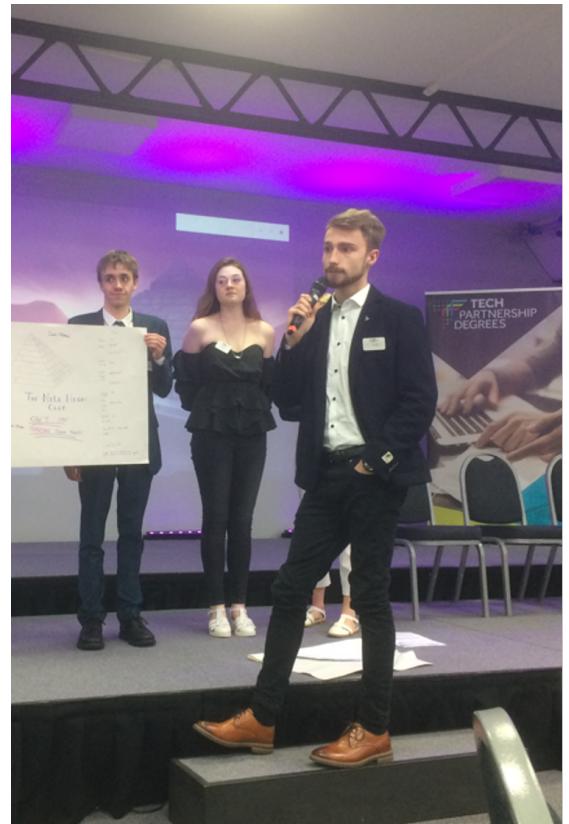
UFCFC3-30-1: Introduction to OO Systems Development

The syllabus will include the following topics:

- Software development lifecycle
- Software development methods (prototyping)
- Problem solving & design with pseudo code (thinking algorithmically)
- Problem solving & Intro to OOA&D with the UML class diagram
- Introduction to a Java IDE (Netbeans)

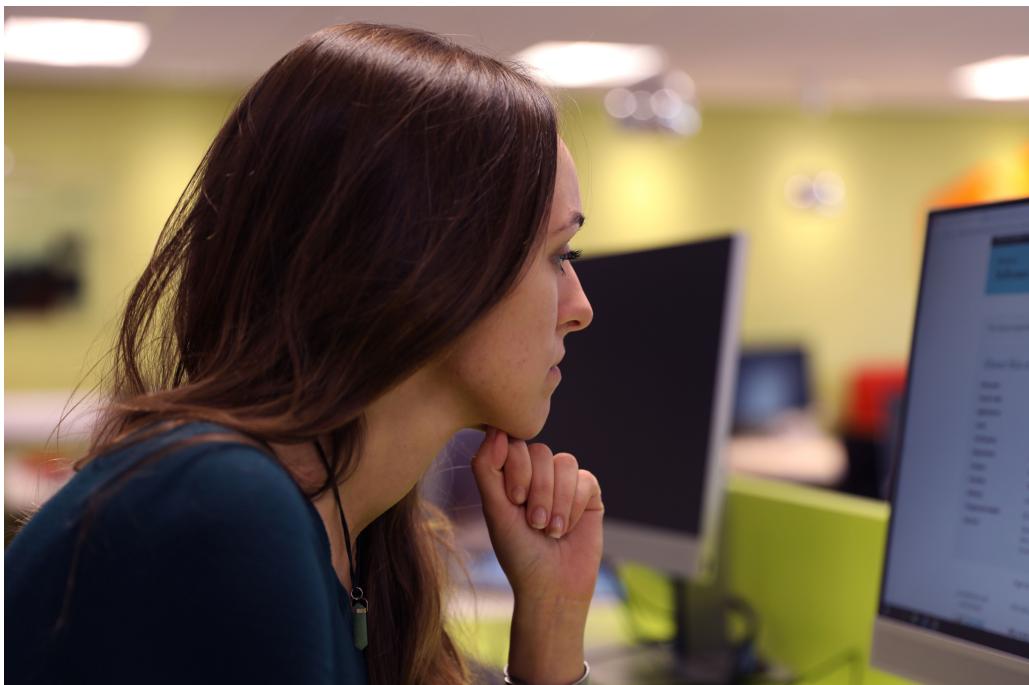
A basic introduction to Object Oriented Paradigm including:

- computer architecture overview
- source code, byte code, machine code, compilers, interpreters
- the role of the JVM
- coding style guidelines
- primitives, classes & objects
- scope of variables
- Iteration & Selection statements
- arrays & collection classes
- file I/O
- interfaces
- inheritance & overriding
- GUIs
- Deploying java applications
- Testing & Use of IDE debugger



UFCFFS-30-1: Foundations of Computing

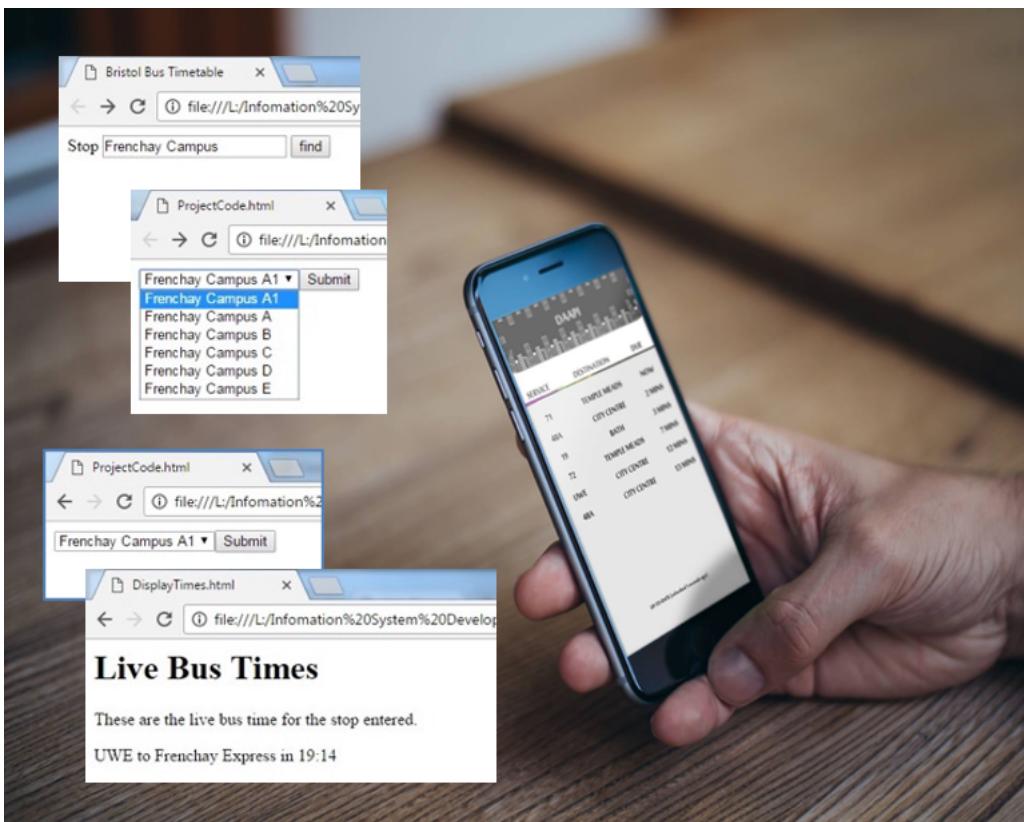
- Computation models: Finite Machines; Pushdown Automata; Turing Machines. How these abstract machines work. What limitations they have. How do apply them to real world applications. The significance of the Universal Turing Machines.
- Formal Languages: words, sentences, languages, grammars, productions. Links to computing models. How to formally define languages. How a compiler detects syntax errors.
- Algorithms: Classes of algorithms, search algorithms and sorting algorithms. Time and space complexity of algorithms. NP-complete problems.
- Recursion: Inductive definitions and recursive programs.
- Logic: Propositional and Predicate logic. Truth tables for basic logic operators. Inference methods.
- Mathematical Structures: Numbers. Sets. Functions. Relations. Matrices. Application of mathematical structure to computing. Enumerating (counting) these structures.
- Graph Theory: Theory and its applications as a modelling tool. Classical problems: finding the shortest route on a graph and the travelling salesman problem.



UFCF9F-30-1: Information Systems Development 1

The course is built around a running example built open transport data. Our approach to data modeling is based on the Unified Modeling Language (UML). Programming practice is developed through the use of Javascript.

- Making the business case.
- Information System Architecture and the Web
- Javascript
- The Systems Development Lifecycle (SDLC)
- Functional Requirements
- Non-functional Requirements
- Requirements Engineering
- Agile: It's all about the code
- Test-Driven Development
- Code Quality Assurance
- Software Configuration Management



UFCF83-30-1: IT Practice: Skills, Models and Methods

This module takes a holistic, integrative approach to human activity systems, information systems and information and communication technology by covering:

- Foundations of information systems practice, methods and tools. Relationships between people, organisations, information systems and information technology. Investigating, analysing and modelling organisations, information systems and IT from different perspectives.
- An introduction to the hybrid role of the information practitioner, identifying information needs of organisations and selecting appropriate technology to satisfy them. The importance of a socio-technical approach in understanding existing and potential information systems.
- Communicating and expressing an understanding of application contexts, information systems and information technology infrastructures, by integrating and applying knowledge from business and technical modules.
- Fundamental concepts of information and communication. Meaning and interpretation in different organisational and social contexts. Information literacy. Search and retrieval. Evaluating information sources.
- Working with people. Forming and managing teams. Communicating, basic record- keeping and documentation. Contemporary examples of projects in information systems.
- Practical communications skills. Report-writing, oral communications, interviewing, giving presentations.
- Learning styles, action learning and reflective practice. Lifelong learning in information systems as a response to social, organisational and technological change.

Year 2 Modules

UFCFN6-30-2: IT Practice: Collaborative Project

This module takes an holistic, integrative approach to information in human activity systems in order to develop students' understanding of the information practitioner through experiential exposure to a wide range of topics from amongst the following:-

- Further understanding of the socio-technical hybrid nature of information practice - modeling and differentiating business, information and technical objectives and benefits
- Understanding and interpreting information activities in workplaces from a user perspective – motivation, participation, user resistance
- Contemporary patterns of IT usage from a management perspective – end- user vs corporate systems, technology and job design, foundations of IT and IS strategy
- Defining, planning, executing, managing and reviewing information projects
- Team-working, team roles, delegation, time management, reporting and accountability.
- Working and communicating with peers, users and business or technical specialists orally, electronically and in writing
- Understanding and questioning assumptions, expectations and opportunities surrounding IT in the workplace from multi-stakeholder perspectives
- Technology, its social context and the search for a good fit between the two
- IS maintenance; introduction to sustainability and information practice
- Development of the information practitioner - using, extending and evaluating methods, techniques, tools and technologies; reflective practice for personal and methodological development



The main assessment is via a team portfolio and presentation with individually monitored contributions assessed at stages during the year incorporating formative advice. This assessment focuses primarily on the practical aspects of the module learning outcomes.

UFCFV4-30-2: Data, Schemas and Applications

The need to store, transform, locate and present data of many forms (numeric, textual, relational, graphical, visual and auditory). Unstructured text through semi-structured networks to homogeneous relational and non-relational structures. Application areas such as personal, corporate, public and collaborative data bases.

- Data Schemas. Separation/Composition of data, schema (including constraints) and meta data and namespaces. Languages for describing schemas - UML models and SQL DDL for tables, data structure diagrams and XML Schema for hierarchical structures. Composition, types and sub-types and inheritance. Data schemas for typical problems.
- Data Manipulation. Standard SQL for relational database management systems (RDBMS) such as MySQL for creation, retrieval, updating and deletion operations. XSLT for XML transformation and JavaScript/PHP for DOM manipulation.
- Web service and communication standards. Application Programming Interfaces (APIs). RSS, ATOM and XML/JSON generating web services. Representational State Transfer (REST). Using data from external services and analysing, reformatting and re-presenting this data. Metadata design, generation and use in the context of the Semantic Web using RDF and SPARQL.
- Scripting technologies and presentation. Server-side processing with PHP including functional programming and MySQL/XML data manipulation. Client side processing with JavaScript and JSON.
- Documentation, technical design and security. Documenting schemas, data structures and interactions. Documenting code including APIs. Optimisation techniques. Testing code and applications. Common data and web application vulnerabilities. Using source code repositories and version management.

UFCFB6-30-2: Object-Oriented Systems Development

- Theory and concepts of the object-oriented paradigm: objects and classes, encapsulation and visibility, cohesion and coupling, instance and class scope attributes and methods, inheritance and polymorphism, leading to abstract classes – overriding and overloading.
- Object-oriented systems software development lifecycles: phases (requirements capture and analysis, design, code, test), iterative and waterfall methods e.g. agile (the agile manifesto), reuse and integration, test-driven development.
- Object-oriented analysis and design techniques: use cases (user stories), verb-noun analysis, abstraction, Class-Responsibility-Collaboration (CRC), and heuristics for design evaluation and test case generation.
- Tools: an UML modelling tool (ArgoUML) with cognitive assistance wizards. A java-based Interactive Development Environment (IDE) e.g. NetBeans, providing application projects, packages, classes, build scripts, deployment, execution, debugging, testing (e.g. JUnit), and version control.
- Software Patterns and Architecture: Model-View-Controller (MVC), analysis patterns (Arlow & Neustadt), design patterns (Gamma et al.), architectural patterns e.g. separation of concerns for multi-tier distributed software systems and interoperability for systems of increasing scale, complexity, multiple interactive channels and virtualisation.
- Persistence: File input / output. Java Database Connectivity (JDBC), serialisation, introductory database entity design and implementation (tables, inserts, queries, result sets etc.)
- Concurrency and Distributed Systems: Theory and potential solutions relating to the changes of concurrency, design and implementation of concurrent systems. Distributed systems – network architectures and protocols, Sockets, Datagram and RMI.
- Graphical User Interfaces: user-centric design, usability testing, event-driven paradigms – graphical components, events, listeners, handlers.

UFCFW4-30-2: Design and Analysis of Data Structures and Algorithms

Review of methods of analysis:

- Mathematical modelling of complexity: space v time.
- Upper and lower bounds: techniques of analysis: correctness, efficiency; analytical strategies, order notation, design strategies.
- Hard problems: the classes P, NP and NP-complete; significance for computing

Classic Algorithms:

- Searching algorithms: linear search; jump searches; worst and average case upper bound determination - probabilistic approaches; randomised searches; binary search; using decision trees to determine the lower bound on search - worst and average cases.
- Selection algorithms: representing rankings using posets; finding maximum and second maximum values; lower bound considerations; finding maximum and minimum; finding i-th largest, randomised and non-randomised approaches.
- Sorting algorithms: strategies for sorting; select sorts; insert sorts; merge sorts; split sorts; analysis of lower bounds and average case.
- Numerical algorithms: exponentials; largest common factor; prime numbers; cryptography; fast Fourier transform.

Data Structures and the algorithms that support and maintain them:

- Collections (sets, dictionaries, hash-tables);
- General sequences (lists);
- Stacks; Queues and Deques;
- Trees (binary, multi-way and some variants such as 2-3-trees, B-trees, KD- Trees and Sphere Trees);
- Graph algorithms: operations on structures; topological sort; depth- and breadth-first search; spanning trees; cheapest paths, travelling salesman problem

Applications:

- Use of language libraries or, where appropriate, handcrafted data structures in application problems such as, parsing arithmetic expressions, tracking the position of mobile units in computer games, providing predictive text and discrete event-based simulation

Optional Placement Year

UFCFE6-15-3: Professional Experience (studied during placement)

This module requires the student to take on a substantial period of supervised work experience, relevant to their academic programme of study, in a professional environment (referred to hereafter as a “placement”).

[NB: the term “placement” does not necessarily mean a single job for the same employer. It may for example involve multiple episodes of work experience with different employers. Therefore where this document refers to the singular terms “employer”, “job”, etc., plural meanings and flexible working modes are also permissible.]

The precise details of the job requirements involved are negotiated and agreed between employer and student, with assistance and advice from the University as needed. Students are generally required to secure an approved placement before taking the module.

Students are expected to demonstrate professional success in working to the employer’s brief, and to reflect critically on the placement in relation to their academic programme. In doing this, they will have exposure to a range of topics from amongst the following:-

- Organisational analysis. Industry-specific trends, developments or emerging issues.
- Contemporary technological changes and opportunities related to their programme/discipline/industry.
- Legal, social, ethical and professional issues (mandatory topic).
- Codes of practice, professional standards, professionalism.
- Ethical practice, sustainability and global awareness.
- Relationships between academe and practice - relate concepts from their academic programme to their professional practice, and vice-versa.
- Reflective practice, lifelong learning and professional development.
- Comparative analysis of different workplaces.
- Personal development planning.

Assessment takes into account the quality of the student’s reflective logbook, employer feedback and coverage of legal, social, ethical and professional issues, and will use published employability measures.

Final Year Modules

UFCFP6-30-3: IT Practice: Consultancy Project

Students are given the opportunity to tackle a complete consultancy project in a live client situation, in consultancy teams working in a self-managed fashion. A wide range of project types is typically involved, including information analysis, requirements engineering, feasibility studies, web design, system development, multimedia design, creative media, music technology, infrastructure development, strategic planning, information systems evaluation, user training and support and process redesign. Projects usually involve external organisations. Consultancy teams are typically expected to:

- negotiate, agree and document a project definition that reflects the business case for change, related explicitly to context, scoped manageably but with sufficient depth and challenge, and demonstrating long-term thinking
- undertake the agreed consultancy activities and produce the agreed deliverables, including sustainability measures and forward planning recommendations for the client
- operate in a substantially self-managed fashion, working within their own project management framework
- delegate tasks and roles to team members in accordance with individual specialisms, interests and needs, and manage individual contributions and quality effectively
- utilise relevant literature, resources and expertise effectively, including previous individual or peer experience as well as staff input
- evaluate and document all aspects of their work appropriately for their client
- review and write up all aspects of their project critically and self-critically for an academic audience

UFCFB5-15-3: Ethical and Professional Issues in Computing and Digital Media

- Role of the professional;
- Role of professional bodies and codes of practice;
- Business, stakeholder and philosophical ethical frameworks;
- Moral and corporate responsibility;
- The Information Commissioners Office; and
- Core: regulation, privacy, security, intellectual property, the public domain.
- Technology innovation (including disruptive technologies) as an important component of change, sustainability and growth.
- How organisations can assess the commercial viability of new technologies and research based ideas and exploit them in their plans and strategies
- Innovative organizational formations enabled by digital technologies

UFCFFF-30-3: Software Development Project

There is no specific syllabus for this module as the project is an individual piece of work, exploring an idea from conception through to realisation. Nonetheless, elements of the project process are covered in a short lecture series at the start of the academic year. The lectures will normally be delivered by the module leader or their nominee. and covers topics such as:

- Choosing a project
- Researching the project idea
- Making use of your supervisor
- Moving from research to requirements
- Configuration and release management and version control
- Quality planning and assurance
- Writing up the project.

In parallel with the lecture series, students will be allocated a supervisor. They will then agree the subject of the project with the supervisor and the Module Leader. Suitable topics may stem from staff, the student, the student's employer or other outside organisations. The topic must lend itself to research followed by a software development process based on the research.

The research component will include the identification of a suitable topic and subsequent investigation from books, papers and other sources.

Requirements should be derived from the research. Software development will include the identification of suitable tools and methodologies to use, giving full and careful attention to the issues of quality and risk.

Students will be expected to demonstrate that they understand the importance of version control as it applies to software and systems; have defined suitable configuration management processes for use throughout the product development lifecycle in storing software deliverables and controlling and tracking changes to their software; have used configuration management and version control tools and release management frameworks and applied sound change management processes properly and effectively when modifying their software designs and deliverables.

Whatever the subject the student will be expected to treat material critically and to demonstrate their understanding of the relevance of material both to their award to the project topic. They will also be expected to reflect on the tools and methodologies used and, at the project completion, comment on their suitability.

UFCFAF-30-3: Development of Information Systems Projects

A review of contemporary approaches to systems development and its methods, tools and practices, frameworks for evaluating methodologies and choosing between them. The components, tools and architectures of complex web-based and workflow systems including business process modelling and enactment, business process management.

- Compare & Contrast Software Development Methodologies / SWEBOK
- Project Management and Planning
- Work Breakdown Structure, LEAN concepts in project management
- Versioning Control and Documentation
- Enterprise Architecture (Zachman framework)
- Introduction to Business Modelling
- Understanding BPMN, Gateways - Simple Tasks
- BPMN Gateways, Events, Tasks, Sub-Processes, Pools, Flows (Camunda)
- BPMN Conversations and Choreographies
- Service Oriented Architectures and Web Services and REST Architecture
- Microservices architecture and DevOps
- Quality Assurance and Testing
- Security and Risks