UNIVERSITY OF
LEADING
THE WAY
WESTMINSTER

PROGRAMME SPECIFICATION

Course record information

Name and level of final award	MSc Software Engineering (Conversion) The MSc Software Engineering (Conversion) is a Master's degree that is Bologna FQ-EHEA first cycle degree or diploma compatible.
Name and level of intermediate awards	Postgraduate Diploma Software Engineering (Conversion) Postgraduate Certificate Software Engineering (Conversion)
Awarding body/institution	University of Westminster
Teaching Institution	University of Westminster
Status of awarding body/institution	Recognised Body
Location of delivery	Cavendish Campus, Central London, United Kingdom
Language of delivery and assessment	English
Mode, length of study and normal starting month	One year full time, two years part time day. September start.
QAA subject benchmarking group(s)	Subject Benchmark Statement: Computing (Master's) 2019. Available at: https://www.qaa.ac.uk/docs/qaa/subject-benchmark-statements/subject-benchmark-statement-computing-(masters).pdf
Professional statutory or regulatory body	Not applicable
Date of course Validation/ Revalidation	July 2020
Date of programme specification approval	December 2020
Valid for cohorts	From 2021/2022
Course Leader	Dr Paul Howells
Course URL	westminster.ac.uk/courses/postgraduate
Westminster course code	PMCSS09F (FT) PMCSS09P (PT)
HECoS code	100374 software engineering
UKPASS code	

1 Admissions Requirements

There are standard minimum entry requirements for all postgraduate courses. Students are advised to check the standard requirements for the most up-to-date information.

For most courses a decision will be made on the basis of your application form alone. However, for some courses the selection process may include an interview to demonstrate your strengths in addition to any formal entry requirements.

More information can be found here: westminster.ac.uk/courses/postgraduate/how-to-apply

2 Recognition of Prior Learning

Applicants with prior certificated or experiential learning at the same level of the qualification for which they wish to apply are advised to visit the following page for further information:

westminster.ac.uk/study/current-students/your-studies/forms-and-procedures/recognition-of-prior-certified-learning-rpcl

3 Aims of the Course

The course is a generalist computing masters programme that has been specifically designed for **non-computing graduates** who:

- have some basic experience and interest in computing and wish to enter the IT industry, or
- who are already within the IT industry and aim to increase their technical skills and knowledge.

Consequently, the course is **not appropriate** for computing/software engineering graduates or graduates where the majority of their degree was computing/software engineering.

Software Engineering is the application of engineering style methods, practices and disciplines to the creation and maintenance of software applications and systems. Hence, a professional Software Engineer when carrying out their work uses these engineering approaches to develop software in a systematic, quantifiable and disciplined way.

The main aim of the course is to provide non-computing graduates with the core leading edge practical knowledge and skills that a professional Software Engineer requires to be successful in today's IT industry. This is achieved by providing students with the essential practical experience of programming and applying software engineering to a number of new and important areas of IT and computing. In addition, the course aims to provide students with the appropriate knowledge underpinning these practical skills.

A key aim of the course is to ensure that graduates have excellent employment prospects in the IT industry. This has been achieved by embedding essential *employability skills* within the course, thus ensuring graduates are fully equipped to pursue a career as professional software engineer.

For example, *technical skills* such as programming, object oriented software development, practical experience of software development tools; *soft skills* such as analytical and critical thinking, communication, presenting and demonstrating, team working, peer reviewing; *career skills* such as CV writing, planning, job searching, interview practice.

The rapid pace of technical change in software development is considerable and this has been accompanied and compounded by an increase in the complexity of the systems that are developed; most noticeably in the dramatic increase in the area of mobile computing. Many applications that run on these systems whether mobile or stationary are distributed in nature and will consume web services provided by service-oriented architectures and cloud-based platforms.

Finally, many people enter the IT industry without a specific educational background in computing, but acquire computing skills and knowledge in the workplace in relatively ad hoc ways.

There is an acknowledged national shortage of IT and computing skills in the workforce, in particular in the specific area of software development.

In response to these issues, the School of Computer Science and Engineering has for many years been running courses that combine an emphasis on methodical approaches to the development of software applications and information systems with a determination to equip software engineering graduates with a portfolio of relevant practical skills and knowledge to allow them to become professional software engineers.

The rationale behind the MSc Software Engineering (Conversion) is to draw on this experience to provide an education that will cover in-depth specific skills and best current practice in software development where there is currently a significant skills shortage, whilst at the same time instilling important research-based skills that will equip students for independent learning in a fast-changing and technically challenging environment.

4 Employment and further study opportunities

Today's organisations need graduates with both good degrees and skills relevant to the workplace, i.e. employability skills. The University of Westminster is committed to developing employable graduates by ensuring that:

- Career development skills are embedded in all courses
- Opportunities for part-time work and work-related learning activities are widely available to students
- Staff continue to widen and strengthen the University's links with employers in all sectors, involving them in curriculum design and encouraging their participation in other aspects of the University's career education and guidance provision
- Staff are provided with up-to-date data on labour market trends and employers' requirements, which will inform the service delivered to students.

The main aim of the course is to provide graduates with the key employability skills they will need to successfully pursue a career in the IT industry, primarily as a Software Engineering, but also in any more general role they may wish to pursue within the industry. The employability skills that have been embedded within the course fall into three main categories:

Technical skills: the main focus for each module is to provide students with core technical computing and software engineering skills in a specific area. For example, object oriented programming, software design, software development within a particular environment, algorithm design and analysis, database systems design and use, mobile and web application development.

Soft skills: all modules require and help students to develop self-direction, critical thinking, effective communication, effective software demonstration, and presentation skills. Modules using group assessments in addition provide students with team working and peer reviewing skills. The project module provides students with project management and research skills.

Career skills: core career management skills are embedded in the course, for example, CV writing, career planning, job searching, self-evaluation and interview practice. These skills are developed within a number of workshops run by the Careers and Employability Service (CES) during the course and are supported online using the University's Engage/Engage Plus career skills development tools. The project module provides students with the support and opportunity to develop career skill, and these are assessed in the module.

The MSc Software Engineering (Conversion) aims to produce graduates who will typically be employed in core software engineering roles across a wide variety of software development environments. Typical job titles within this area include Software Engineer, Systems Administrator, Programmer, Mobile Application Programmer, Software Designer, Web Developer, Web Application Programmer. Many of these jobs require a knowledge of computer systems ranging from general skills in designing and programming, to specialized mobile and/or web applications programming. Their common theme is that they require the ability to understand and competently apply the various techniques related to the stages of the “software life-cycle" process.

In addition to these core software engineering roles, graduates would be able to find employment in a wider range of IT roles by combining their existing skills and experience with the new skills acquired on the course. For example, Business graduates would have the opportunity to join Technical Consulting, Sales Engineering, Technical Program Managers roles.

Graduates would also have a sound understanding and knowledge required to undertake a specialist masters degree in a software engineering area, or a research degree related to the object-oriented paradigm.

5 Course learning outcomes

Learning outcomes are statements on what successful students have achieved as the result of learning. These are threshold statements of achievement and are linked to the knowledge and understanding (KU), specific skills (SS) and key transferable skills (KTS), that a student will have gained on successfully completing a course.

On successful completion of the course, students will be able to:

(CLO1) Select and apply the object oriented techniques appropriate to develop and construct a software system by following the stages of the software life-cycle: requirements analysis, design, implementation, testing and maintenance. (SS, KU)

(CLO2) Critically evaluate the object oriented paradigm and its role at various stages of the software development life-cycle. (KU)

(CLO3) Critically evaluate and work within a range of typical software engineering environments by selecting and using the appropriate software development tools and techniques for ensuring software quality. (SS, KU)

(CLO4) Explain the role of the software engineer in a wider context and work within the relevant professional, legal and ethical frameworks within the software engineering profession. (KU)

(CLO5) Analyse and explain the complexity of software engineering issues, both systematically and creatively, and to formulate sound judgements in the absence of complete information. (SS, KU)

(CLO6) Demonstrate self-direction in researching, recognizing and tackling problems, and to act autonomously, and where necessary with originality, in planning and undertaking their activities to a professional standard. (KTS)

(CLO7) Analyse, interpret and explore technical and theoretical knowledge and practical experience at a specialist level within software engineering. (KU)

(CLO8) Critically evaluate the data processing requirements of real world problems, select and design efficient data models and algorithms for a software solution that satisfies the problem's requirements. (SS, KU)

(CLO9) Critically evaluate different data modelling approaches and techniques to design and construct fit for purpose data repositories for capturing, storing and processing large volumes of data. (SS, KU)

(CLO10) Develop as self-critical learners by independently evaluating their own and others work, recognising their development needs and satisfying them by acquiring new skills and knowledge to a high standard. (KTS)

(CLO11) Work effectively within a team both as a leader and/or member, clarify tasks and guide the activities of others, make appropriate use of team members abilities, negotiate and handle conflict with confidence, and participate effectively in the peer review process to improve practice and outcomes. (KTS)

(CLO12) Communicate confidently and effectively in academic and professional environments using written media, orally and software demonstrations. (KTS)

6 Learning, teaching and assessment methods

The guiding principles are to provide sufficient and appropriate teaching facilities, learning resources and student support services to deliver a high-quality academic experience. In practice, this means providing and maintaining appropriate physical and virtual environments for delivery of learning and teaching; learning facilities and resources are accessible and relevant to students' development of their knowledge and skills; actively encourage and monitor student engagement individually and collectively in their experience and effectiveness of the learning and teaching provision; support all students to achieve successful academic outcomes; design and deliver learning and teaching to develop subject-specific and transferable skills that enhance students' personal development and employment opportunities.

The specific teaching and learning strategies adopted on the course use a variety of inclusive learning, teaching and assessment methods to ensure that together they enable and empower every student to fulfil their potential and achieve a successful outcome. The wide range of necessary knowledge and skills required for the successful development of complex software systems means that a correspondingly wide range of strategies is adopted to facilitate students in their acquisition of this knowledge and development of these skills. For example, material is presented effectively to students in ways that emphasise the combination and blending of both the theory and practical nature of the subject.

Supporting the University of Westminster's Black Lives Matter Commitment Plan, the course has been developed using an inclusive approach where students will have a learning experience that respects diversity, encourages their participation, reduces barriers to learning and considers the varying needs of students. Inclusivity has been addressed by welcoming any candidate with interest and basic experience in computing along with a good degree in any non-computing subject, for example, the arts, humanities, sciences, etc. Within the course, inclusivity has been addressed through a programme that offers a wide range of: software engineering topics blending both practice and theory, different learning and teaching methods, diverse assessment methods, personalised learning through accessible online resources, individual support throughout the course and opportunities for professional development. The topics covered in the course focus entirely on essential technical and soft skills, but where appropriate will be studied from different social and cultural contexts providing students with a more robust and rounded approach to the subject. For example, a diverse range of case studies, authors, reading lists and critical perspectives are embedded within the course to provide students with an inclusively designed and diversified curriculum. Together these encourage and enable students to be active, fully participate and tailor their learning according to their career aspirations and individual needs.

6.1 Learning methods

Learning methods are aimed at facilitating a student's active and critical learning by the acquisition, understanding and application of knowledge, skills and professionalism. The learning methods employed on the course vary depending on the type and content of a module and its intended learning outcomes. Consequently, a wide range of learning methods are used across the course's modules, for example, the use of:

- specialised software tools and packages, such as Software Development Environments and Computer Aided Software Engineering (CASE) tools, to build students hands on skills and understanding of such tools;
- case studies, to improve students' analytical and problem-solving skills; moreover, to integrate the knowledge gained in individual modules and demonstrate how the accumulated knowledge and understanding can be used, common case studies, where possible, are used across modules, with each module tackling different aspects of the same problem;
- presentations from outside speakers with industrial experience, where appropriate, to enable students see how the taught material is applied in industry; appreciate how industry uses the various technologies / tools / methods / techniques to produce solutions;
- team/group work, to enable students to develop further their teamwork skills to work effectively in a professional environment;
- research methods involving the use of library and online sources to develop students research and analysis skills;
- academic report writing as part of the assignments set, to develop further these important skills, including those related to formatting and proper use of referencing;
- presentation and seminar sessions during which students present work to their classmates and evaluate/assess each other's work;
- continuous encouragement to exploit networking opportunities and to participate and get involved in community organised events, as these enable students to identity areas for improvement while demonstrating their skills and knowledge on specific subjects / topics;
- work based projects, where appropriate, to enable students to (a) participate in a real life project; (b) develop/enhance a high degree of organisational skills as they have to work under the guidance of industrial supervisors and adhere to strict timetables for deliverables and deadlines; (c) be exposed to work ethics and culture, and experience the ethos of a workplace environment all of which enhance further students' social, interpersonal and professional skills;
- assessment and feedback as an integral part of the learning process to enable students to (a) gauge their progress in relation to learning outcomes; (b) reflect on what they have learned; (c) identify areas in which they are strong and areas in which they need to improve their learning so that students develop the rights skills to enable them to achieve the required learning outcomes; and (d) help them make informed decisions on the pace and focus of their own independent learning;
- assessments as a tool to develop/enhance students' skills and competences; for example working on an assignment as part of a team will help students develop/practice their group working skills, whereas an essay and research report can be used not only to assess students' knowledge and understanding of a specific topic(s), but also help develop students' academic writing skills.

Students are supported throughout their studies by *Blackboard*, the University's Virtual Learning Environment (VLE), web-based teaching materials and the Library and IT services. Blackboard provides access to sites that provide important information related to the course, individual modules, and general university information. The Blackboard module sites are used as repositories for lecture notes, online reading lists, tutorials exercises, lectures and tutorials recordings, assessment schedules, coursework (including feedback) and for assessment purposes, e.g. online tests. The course recognises the importance of individuals being able to function equally well both as individuals and as members of a team; thus, group activities are encouraged and promoted. To support and encourage student face to face interaction and collaborative work through exchange of emails, files, and online discussions, the facilities offered by Blackboard and other online systems are commonly utilised. The University also provides other online support especially important for students on the course, these include *AppsAnywhere* that allows access to University applications on student's computers whether on or off the University sites, and *ServiceDesk that provides* online IT support.

6.2 Teaching methods

Teaching methods are aimed at encouraging and involving the active participation of the students in their learning through knowledge and skills acquisition by means of engagement, experimentation, self-study and practical experience. The teaching strategies employed on the course are wide ranging and vary across the modules that make up the programme of study. Those selected for an individual module depend on what is most appropriate for the module's topic, learning outcomes and assessment strategy.

The delivery of the course's taught modules involves using lectures, tutorials, workshops and seminars. The lectures are used to provide a firm grounding in the theory, methods and techniques relevant to the module’s topic. Within lectures a range of approaches are adopted, such as, traditional lectures, and ‘structured lectures’, where lecturing is broken up by periods of student-led activity. Lectures are usually supplemented by instructor led sessions where a more experimental, investigative and problem-solving approach is adopted, than is feasible in a formal lecture, to solve theoretical and/or practical problems. During these sessions students will attend problem solving tutorials or workshops, where they work at their own pace, working alone or in small groups, occasionally working on paper, but usually working at a PC or workstation, always with a member of staff guiding the work or on hand to help resolve problems. To integrate the knowledge gained in individual modules common case studies, where possible, are used across modules, with each module tackling different aspects of the same problem. Modules with a highly technical and practical content are typically delivered in the form of workshops. These take place in a computing lab and they combine material normally covered in a lecture with practical/hands-on exercises. In particular, the various concepts/constructs of the module’s topics are introduced in short bursts and they are followed by a series of practical exercises that aim at enabling students to appreciate these and understand how they can be used. This approach encourages students to actively participate in the development of a solution by allowing them to (a) express their thoughts; and (b) get immediate individual feedback from peers and/or the instructor. Finally, there are also seminar sessions in which students will present work to their classmates and assess each other’s work.

The project is the most important aspect of the Master’s programme, as it plays the key role of unifying and integrating the material taught on the course's modules and provides the opportunity to put into practice and extend what has been learnt in a real world situation. Students are expected to work on a software development project that is in an area chosen by the student. For these reason different teaching methods are used to deliver the project module. To help students successfully complete the project all students are allocated a supervisor who is a member of academic staff. The project supervisor allocated to a student will, in most cases, have research interests in the area of a student’s chosen project area. The initial role of the supervisor is to advise and guide a student in their choice of project topic, this ensures that it is an appropriate project topic at the level and standard expected. Students are also encouraged to consult other members of staff as part of this process. In general, the supervisor's role is to act as someone who will guide students throughout the various phases of the project and to whom students will discuss their project work and receive feedback from. For example, the progress made, relevant technical and research matters, advice on planning and future activities. Supervisors will also help students (a) decide on the scope of the project; (b) devise a project plan; (c) monitor their progress and adhere to target dates for assessments; and (d) writing up the final project report. In addition, students are supported by a series of seminars and workshops that aim to equip them with the required background knowledge and skill required for the project by covering the selection of a project topic, managing a project, the main project activities, researching their project area; and devising a project proposal sufficiently detailed to enable them to complete their project to a successful outcome.

The course supports students at all stages through the course, in particular with the transition into becoming a skilled programmer and software engineer. Support is provided before and during the course. Before students commence the course they will be provided with introductory material intended to prepare them for their initial modules. This will be built on during the course induction by workshops that review the introductory material and introduce the main software and computer systems they will be using on the course. Once module teaching begins students will be supported by lectures, workshops and practical tutorials. In addition, each module will provide the following online support: recording of lectures, access to teaching material in a range of formats, online reading lists and access to e-books, discussion boards, virtual study rooms for students to collaborate, individual and group online meetings. Individual support for each module will be available from the module's teaching staff. At key stages in a student's academic studies the decisions they will need to make, e.g. choice of option modules and project topic, will be guided and supported by the course leader and other members of the course team, as well as their personal tutor. Students will also be supported and help with personal issues that arise during their studies by their personal tutor, course leader and senior tutor.

6.3 Assessment methods

Assessments and feedback are important and are designed to form part of the learning experience and they can have a variety of types and forms. For example, assessments may involve practical exercises ranging from small tasks that might be completed in a tutorial, to more complex tasks, like the design and creation of an artifact, e.g. software, or the investigation/research on a topic/area. Some of the assessments are designed to be completed individually, whereas other assessments may require students to work as part of a team, emulating as close as possible the environment students will face in later life in industry. Types of assessment used in the course include essays, technical / lab reports, practical tests/exercises, quizzes, in-class or online tests, practical exercises, portfolios, demonstrations, oral presentations, vivas, project reports, time constraint examinations, etc.

Assessment can be (a) formative (i.e. helps establish where students are in their learning and what they have learned so far), or (b) summative (i.e. measures how much they have learned in a way that contributes to their overall grades). The type and nature of the employed assessment methods varies depending on the module and its associated learning outcomes. The guiding principles in designing/choosing a module’s assessment and its associated feedback include:

- the choice of assessment method(s) employed needs to provides an opportunity for new learning and contribute to the learning process;
- the assessment method used should be fit-for-purpose able to measure students' achievement in the module's associated learning outcomes of each module;
- assessment is criterion-based, i.e. assessed work is marked using clearly stated assessment criteria,
- in selecting assessment methods consideration is given to the amount of effort and time required to complete the task(s) and to maintain an acceptable and balance assessment loading;
- timely and formative feedback is to be given for all assessments, including examinations;
- providing information about how the student has performed in the (summative or formative) assessment;
- guidance on how the student can improve their performance in future, either individually or as part of a team.
- the feedback students will receive needs to enable students to make that transition from small practical exercises to more complex pieces of work towards your postgraduate project.

All assessments that contribute to final grades will be assessed against clear assessment criteria stated in module descriptors; these assessment criteria are directly linked to the module's learning outcomes and they will be used to evaluate the submitted work and produce written feedback. Marks will be produced following rigorous quality mechanisms that ensure academic judgement remains fair and consistent with the wider educational sector. Feedback is given in various forms and stages; for example, in response to assessment, in response to questions in lectures, seminars and tutorials, and in guidance given during the supervision of student projects. Feedback will also come from interactions with other students.

The assessment diet of most of the modules involves a mixture of practical coursework and a closed book problem solving focused examination. For most of the modules, the coursework component involves a few assessment elements that may involve laboratory work, technical reports, oral presentations, in-class (written or online) tests, etc. The project, which is a substantial piece of work that involves the investigation/research of a topic and the development of software, is assessed using a written project proposal, final report and a viva where the students need to discuss and defend their work and findings, and demonstrate their software.

7 Course Structure

This section shows the core and option modules available as part of the course and their credit value. Full-time Postgraduate students study 180 credits per year.

Credit Level 7				
Module Code	**Module Title**	**Status**	**UK Credit**	**ECTS**
7SENG011W	Object Oriented Programming	Core	20	10
7SENG010W	Data Structures and Algorithms	Core	20	10
7SENG012W	Software Development Environments	Core	20	10
7BDIN007W	Data Repositories Principles and Tools	Core	20	10
7SENG003W	Advanced Software Design	Core	20	10
7SENG013W	Software Development Project	Core	40	20
7SENG014W	Web Application Development	Option	20	10
7SENG002W	Mobile Application Development	Option	20	10
7CSEF002W	Cyber Security Threats and Countermeasures	Option	20	10
7BUIS009W	Data Visualisation and Dashboarding	Option	20	10
7BUIS010W	Data Warehousing and Business Intelligence	Option	20	10
7BDIN006W	Big Data Theory and Practice	Option	20	10
7BUIS025W	Web and Social Media Analytics	Option	20	10
	Other School Master's Module in consultation with the Course Leader	Option	20	10

Please note that not all option modules will necessarily be offered in any one year. In addition, timetabling and limited spaces may mean you cannot do your first choice of modules.

7.1 Part time students

Part time students are normally expected to complete the course over a two-year period, see table below. This means that part time students cover the taught part of the course over four semesters (two years) and that they do their project in their second (final) year of study, completing it during the summer months of that year.

Year of Study	Module Code	Module Title
1	7SENG011W	Object Oriented Programming

1	7SENG012W	Software Development Environments
1	7SENG010W	Data Structures and Algorithms
1	7SENG003W	Advanced Software Design
2	7BDIN007W	Data Repositories Principles and Tools
2	7SENG013W	Software Development Project
2		Option Module 1
2		Option Module 2

8 Professional Body Accreditation or other external references

Currently the MSc Software Engineering (Conversion) course does not have any Professional Body Accreditation.

9 Academic Regulations

The current Handbook of Academic Regulations is available at westminster.ac.uk/academic-regulations. In some cases course specific regulations may be applicable.

10 How will you be supported in your studies?

10.1 Course Management

The management structure supporting the course is as follows:

- Course leader: responsible for the running and overall management of the course and development of the curriculum.
- Module Leader: responsible for overall management of the module, coordinating the module team and for the delivery, resourcing and smooth running of the module.
- Course Team: comprises the Course Leader and all the members of staff who teach on the course.
- Personal Tutor: responsible for providing academic and personal support for a student throughout their studies.
- Head of School of Computer Science and Engineering, holds academic responsibility for the course, and for the other courses within the School within the College of Design, Creative and Digital Industries.
- Head of the College of Design, Creative and Digital Industries, holds overall responsibility for the course and for other courses run by the College.

10.2 Academic Support

Upon arrival, an induction programme will introduce you to the staff responsible for the course, the campus on which you will be studying, the Library and IT facilities, additional support available and to your Campus Registry. You will be provided with the Course Handbook, which provides detailed information about the course. Each course has a course leader or Director of Studies. All students enrolled on a full-time course and part time students registered for more than 60 credits a year have a personal tutor, who provides advice and guidance on academic matters. The University uses a Virtual Learning Environment called Blackboard where students access their course materials, and can communicate and collaborate with staff and other students. Further information on Blackboard can be found at westminster.ac.uk/blackboard.

10.3 Learning Support

The Academic Learning Development Centre supports students in developing the skills required for higher education. As well as online resources in Blackboard, students have the opportunity to attend Study Skills workshops and one to one appointments. Further information on the Academic Learning Development Centre can be found at westminster.ac.uk/study/academic-learning-development

Learning support includes four libraries, each holding a collection of resources related to the subjects taught at that site. Students[1] can search the entire library collection online through the Library Search service to find and reserve printed books, and access electronic resources (databases, e-journals, e-books). Students can choose to study in the libraries, which have areas for silent and group study, desktop computers, laptops for loan, photocopying and printing services. They can also choose from several computer rooms at each campus where desktop computers are available with the general and specialist software that supports the courses taught in their College. Students can also securely connect their own laptops and mobile devices to the University wireless network.

10.4 Support Services

The University of Westminster Student and Academic Services department provide advice and guidance on accommodation, financial and legal matters, personal counselling, health and disability issues, careers, specialist advice for international students and the chaplaincy providing multi-faith guidance. Further information on the advice available to students can be found at westminster.ac.uk/student-advice.

The University of Westminster Students' Union also provides a range of facilities to support students during their time at the University. Further information on UWSU can be found at westminster.ac.uk/students-union.

11 How do we ensure the quality of our courses and continuous improvement?

The course was initially approved by a University Validation Panel in 2020. The panel included internal peers from the University, academic(s) from another university and a representative from industry. This helps to ensure the comparability of the course to those offered in other universities and the relevance to employers.

[1] Students enrolled at Collaborative partners may have differing access due to licence agreements.

The course is also monitored each year by the College to ensure it is running effectively and that issues which might affect the student experience have been appropriately addressed. Staff will consider evidence about the course, including evidence of student achievement, reports from external examiners in order to evaluate the effectiveness of the course.

A Course revalidation takes place periodically to ensure that the curriculum is up-to-date and that the skills gained on the course continue to be relevant to employers. Students meet with Revalidation panels to provide feedback on their experiences. Student feedback from previous years is also part of the evidence used to assess how the course has been running.

12 How do we act on student feedback?

Student feedback is important to the University and student views are taken seriously. Student feedback is gathered in a variety of ways.

- Through student engagement activities at Course/Module level, students have the opportunity to express their voice in the running of their course. Student representatives are elected to expressly represent the views of their peers. The University and the Students' Union work together to provide a full induction to the role of the student representatives.
- There are also School Staff Student Exchange meetings that enable wider discussions across the School. Student representatives are also represented on key College and University committees.
- All students are invited to complete a questionnaire before the end of each module. The feedback from this will inform the module leader on the effectiveness of the module and highlight areas that could be enhanced.
- The University also has an annual Postgraduate Taught Experience Survey or PTES which helps us compare how we are doing with other institutions, to make changes that will improve what we do in future and to keep doing the things that you value.

Please note: This programme specification provides a concise summary of the main features of the course and the learning outcomes that a student might reasonably be expected to achieve and demonstrate if s/he takes full advantage of the learning opportunities that are provided. This specification should be read in conjunction with the Course Handbook provided to students and Module Handbooks, which provide more detailed information on the specific learning outcomes, content, teaching, learning and assessment methods for each module.