Unit Specification

|  |
| --- |
| UNIT SPECIFICATION *Please note: the following information will be displayed in the publicly available unit catalogue* |
| Department or School – and for new units the department/school code Please note that the code selected will determine the first 4 characters of the code in SITS, please see - <http://www.bristol.ac.uk/academic-quality/approve/onlineapprovaltool/sitscodes.html> |
| Engineering mathematics and technology |
| Unit Code (revised units only) |
|  |
| Unit Title (not editable unless setting up a new unit) |
| Introduction to Computer Programming and Algorithms |
| Credit Points (not editable unless setting up a new unit) 1 credit point normally equals 10 hours total student input (including study, revision and exams) |
| 20 |
| Level of Study (not editable unless setting up a new unit) For further information on levels please see [national level descriptors](http://www.bristol.ac.uk/academic-quality/approve/approvalguidance/faqs/levels.html/?_ga=2.13641407.908273768.1537779455-1438882111.1528284054) |
| UG Yr1 |
| Academic year (new units only) 1st academic year of entry for students, e.g. 2021/22 |
| N/A |
| Unit Director A Unit Director must be a member of University member of staff. For further guidance for Unit Directors please see: <http://www.bristol.ac.uk/media-library/sites/academic-quality/documents/unit-director-role-spec.pdf> |
| Hemma Philamore and Martin Garrad |
| Open Unit Status Please state if the unit is an open unit, as defined in the open units scheme at:[www.bristol.ac.uk/academic-quality/approve/openunits/](http://www.bristol.ac.uk/academic-quality/approve/openunits/). *From 21/22 this will only be editable by AQPO.* |
| No |
| Unit Information (previously called Unit description) Include the following two headers in this section: Why is this unit important? [150 words] This is the first thing students will read. Inspire them! Give them a sense of why it will be invaluable for them to study it. For mandatory units: explain the rationale for its status; for optional units: this is the text you will use to recruit students to the unit. How does this unit fit into your programme of study [100 words] Give a sense of the context within the overarching narrative of the programme. NB that a unit may fulfil different purposes in different programmes, and this should be made clear (e.g. in Single Honours programmes the unit covers a core disciplinary competency, as an option in other programmes it complements core units or allows students to appreciate further contexts, etc.). If the unit is must pass on some or all programmes, make this clear. |
| Why is this unit important?  The world runs on algorithms- from your Netflix recommendations to a bank deciding whether you should be approved for a mortgage (or not!). Within an engineering context, algorithms enable computer aided design tools, allow simulation of complex physical problems, and underpin AI tools such as ChatGPT or DALL-E; being able to think algorithmically is a core skill for the 21st century engineer. Closely connected to algorithmic thinking is knowledge of computer programming, which will enable you to translate algorithms into instructions that can be executed by a computer to solve real world problems. Programming is also a requirement for employments in areas such as software engineering, data science, AI and robotics. Finally, learning to code is fun!  How does this unit fit into your programme of study  You will use programming in follow-on units including mathematics and data modelling and scientific computing. A basic knowledge of algorithms is essential for understanding more advanced topics in machine learning and AI. |
| Units you must take before you take this one (pre-requisite units) Please list all units which must be taken *before* starting this unit or state ‘none’. |
| None |
| Units you must take alongside this one (co-requisite units) Please list all units which must be taken *alongside* this unit or state ‘none’. |
| None |
| Units you must not take alongside this one Please list all units which students must not take or state ‘none’. |
| None |
| Your learning on this unit (previously called Intended Learning Outcomes) Include the following three headers in this section [250 words]: An overview of contentHow will students, personally, be different as a result of the unit what you know, how you think and what you can do. Learning Outcomes distinct from what students will learn. Keep them direct, student-facing and concise – no more than three or four. Where possible link outcomes to the [Bristol Skills Framework](https://www.ole.bris.ac.uk/bbcswebdav/courses/Study_Skills_2016/PDP/Skills%20framework/index.html#/id/5f3d46bd02b59503800af557;). Highlight any features which prepare students for employability or other skills which students may deploy after graduation. See [guidance on producing learning outcomes](https://www.bristol.ac.uk/academic-quality/approve/approvalguidance/intendedlearningoutcomes/) |
| An overview of content On this unit, you will learn:   * The fundamentals of programming (data types, controlling the flow of a computer program) * How to read, write, process and visualize data * How to write code that is not only correct, but also able to understood by others * How to understand and implement some basic algorithms (e.g. for searching or sorting) * How to translate a mathematical model or algorithm into a computer program   We will learn using Python, which has the following benefits:   * Free and open source * Versatile - both a scripting and a programming language:   + scripting : run in a host application for debugging and viewing output (e.g. a mathematical model)   + programming : controls a computer or machine (e.g. a microcontroller on a robot) * Increasingly used in industry and academia * Large and diverse community of users and developers  How will students, personally, be different as a result of the unit On successful completion of this unit, students will be able to:   * Apply the process of designing, writing and debugging a program. * Move from a problem statement to a computational formulation of an algorithm for solving a practical problem. * Use and understand basic computer programming terminology. * Understand and implement widely used algorithms in code * Employ good programming practice in the creation of computer programs. * Use basic procedural programming constructs appropriately, with correct syntax.   **Learning outcomes**  At the end of this course, students will be able to:   1. Describe the basic principles, concepts and terminology used in computer programming. 2. Apply the fundamental tools of computer programming to write simple computer programs. 3. Translate high-level "problem statements" into simple algorithms, and implement these algorithms as a computer program. |
| How you will be assessed (previously called Assessment Information) Include the following three headers in this section: Tasks which help you learn and prepare you for summative tasks (formative): Give a brief rationale for the formative, indicating how it links to the summative. Include details of specific formative assessment points – what will students have to do and what kind of feedback can they expect. Include information about the timing of these tasks. Tasks which count towards your unit mark (summative): Include brief details of any summative assessment task(s). If there are special features to the assessment, e.g. some elements are aggregated whilst others are must pass/ pass mark differs from university standard, explain clearly here. Include information about the timing of these assessments.  Consider the rationale for the overall volume of assessment, and for the ratio of formative to summative. When assessment does not go to plan This section outlines re-assessment arrangements for students who have not been able to take or pass a summative assessment. The re-assessment task must demonstrate achievement of the learning outcomes covered by the original assessment task. Any special arrangements for referrals must also be clearly stated here e.g.: Will the re-assessment task take a different form to the original assessment? If there is Group assessment what arrangements will be made for the reassessment of group work – in particular, how will individuals who failed a group-work assessment be re-assessed? If student numbers permit, will a new group be constituted by the unit leader – if so, how will that new group be supported? Or will individual/reflective work be set? |
| Tasks which help you learn and prepare you for summative tasks (formative) In TB1, weekly lab exercises (asking you to demonstrate programming fundamentals) will be released. These can be submitted at any time during weeks 1-11 for automatic marking. These exercises are identical to the exercises used for the first summative task.  Throughout the unit, programming fundamentals will be put into practice in a number (4-6) of consolidation activities that require you to integrate multiple fundamentals to solve a problem. Code produced for these activities will be submitted and class level feedback provided. Consolidation activities will be similar in scope to the second summative task, while feedback will familiarise you with the process, we will use for assessing the quality of the code you produce.  Towards the mid-point of TB2, one consolidation activity will be submitted as a group, with each group receiving an in-class mock viva. This will familiarise you with the kinds of questions you may be asked during the assessment for the second summative task. Tasks which count towards your unit mark (summative) There will be two assessments. At the end of TB1, you will submit a code portfolio which will comprise a subset of the weekly lab exercises (30% of final grade). This is to test your ability to apply the fundamental building blocks of programming (ILO 2). You will also submit a coursework assignment (70% of final grade) comprising a modelling exercise and report, which will be assessed by 100% viva. This is to test your ability to produce a complete program and explain how it works (ILO 1 and 3). When assessment does not go to plan Reassessment will involve redoing and resubmitting the assignment including a second viva |
| How you will learn (previously called Teaching Information) Help students not just to be clear about the different learning activities they will do and understand the learning approach (student-centred; interactive; applied; inquiry-based; reflective; linked to industry; problem-based etc.) but also why these ways of learning are best suited to the unit, the learning outcomes and the assessment choices. Include here formative exercises which do not contribute to the summative grades |
| Learning will be predominantly through application of programming and algorithmic principles; we believe the best way to learn to code is to write code (ILO 2, 3). This learning will be supported by activities that require you to read, understand, reflect, extend, and correct code written by yourself and others. These activities will support your ability to design effective computer programs and communicate how and why you have chosen to write a piece of code (ILO 1, 3).  You will be required to pre-watch/pre-read provided study material about fundamental programming and algorithms. You will attend weekly lab sessions where you will work through an exercise sheet to gets hand on experience of writing computer code. These sessions will alternate between introducing fundamental programming concepts, and applying these concepts together to solve simple problems.  Every two weeks you will also attend a seminar where we will discuss and engage in group work on high level concepts of programming and algorithms. |
| Reading and References This section will be replaced from 21/22 with a piece of text linking to the University’s Resource Lists. As all sections must be completed, please state n/a here. **For new units** please provide an indicative resource list (or link to draft Talis resource list) in the approval form section. |
| Teaching Block State multiple teaching blocks if this unit runs more than once per year. If the teaching block is changing please state which block it was and which it is moving to. |
| TB4 |