UNIT 4: PROGRAMMING

Delivery guidance

Approaching the unit

This internally assessed unit has been designed to give your learners the opportunity to explore the concepts of computer programming as well as its implications and applications in a vocational context. You will be teaching learners to apply analytical thinking skills by considering the use of different programming paradigms to solve problems and create solutions. By the end of the unit, learners should be able to consider the needs of a client and develop solutions to meet a set of identified requirements.

This delivery guide does not cover everything that needs to be delivered for completion of this unit but gives examples of delivery methods. You should refer to the specification for full details of all the content that needs to be covered.

Delivering the learning aims

For learning aim A, you could start by exploring the concepts of computational thinking. Initiate a class discussion about how these concepts are applied to the process of problem-solving and computer programming. Put learners into small groups and ask them to examine how to apply computational thinking skills to a given concept or problem. Learners should consider how different software is used to solve different problems. You should give learners a wide range of example programs, code bases and problems to investigate, to which they will need to apply analytical and computational thinking skills. For instance, learners could start by identifying common trends in a particular type of program and the ways in which these needs are met. Learners should explore different programming paradigms, in high-level and low-level programming languages, and how they are used to solve a range of problems. You could start by examining the principles of each paradigm, the functions that it offers and its benefits and drawbacks. It would be beneficial for learners to explore actual code that follows a specific paradigm. Learners should be given the opportunity to explore, change and develop the given code in order to see the implications of effective and less effective code.

Allow time in your sessions for learners to explore common programming constructs, functions and logical and mathematical principles. It may be beneficial to explore these in general terms using pseudocode before applying them to different languages and paradigms. Ask learners to compare the performance of different programming languages, programming techniques and different types of program/solution using testing methodologies and benchmarking tools. Give them access to examples of programs and code bases written in a range of languages. Open-source projects and communities could be good sources for appropriate material.

For learning aim B, you will need to ensure that learners have a sound understanding of the application of the principles and process of the software development lifecycle. Learners are not required to follow a particular model, although they may need to discuss the processes that they choose to use when evaluating their performance and solution.

You will need to help learners to develop practical project planning and management skills in a vocational scenario. When preparing for the assignment, you should ensure that learners are familiar with producing clear documentation and effective plans for all stages of developing a software solution. Learners should be able to communicate ideas effectively and be aware of the conventions and expectations of communication in professional, vocational scenarios.

Before starting the assignment, learners will need to have a good understanding of programming principles and their application in different programming paradigms. Ensure that learners are given the opportunity to practise creating different programs to meet a range of identified needs before attempting the final assignment. Ideally, learners should practise applying a range of different programming languages so that they are able to effectively select and apply a suitable paradigm to solve the problem presented in the assignment.

Learning aim C requires learners to demonstrate appropriate testing and review methodologies. You should ensure that learners have a good understanding of selecting and applying different testing methods, creating and completing test documentation and working with others to review and refine solutions. There is an opportunity for learners to test each other's solutions as users and/or as critical technical experts. Learners should be able to review the success of a solution, and their own performance, against the project success criteria. When reviewing the quality of outcomes, encourage learners to draw upon a range of sources including feedback from others. Learner can make use of the outcomes of testing and the notes that they have made throughout the development that give details of the process.

Throughout their practical work, learners should be encouraged to keep a diary, in which they can keep a record of their progress, any issues they encountered and how they overcame them. This will be valuable when writing the evaluation and reflecting on their own performance as part of the second assignment.

Accurate communication skills are vital for progressing to higher education and in employment. Learners should be confident in presenting thoughts and ideas to others, as well as producing well-presented, accurate and appropriate documentation for all stages of a project. Learners must be able to effectively evaluate the success of a project and the factors that contributed to the final outcomes, including their own skill, knowledge and behaviours.

Learning aim	Key content areas	Recommended assessment approach
A Examine the computational thinking skills and principles of computer programming	**A1** Computational thinking skills **A2** Uses of software applications **A3** Features and characteristics of programming languages **A4** Constructs and techniques and their implementation in different languages **A5** Principles of logic applied to program design **A6** Quality of software applications	A report evaluating computational thinking skills and how the principles of software design and computer programming are applied to create effective, high-quality software applications
B Design a software solution to meet client requirements	**B1** Software development life cycle **B2** Software solutions design	A project brief identifying the scope of the problem and user/client requirements Design documentation for the suggested solution User feedback and design refinement documentation Development and support documentation, including development and testing logs, meeting notes and a report that evaluates the outcomes and development of the project
C Develop a software solution to meet client requirements	**C1** Software solutions development **C2** Testing software solutions **C3** Improvement, refinement and optimisation of software applications **C4** Review of software solutions **C5** Skills, knowledge and behaviours	

Assessment guidance

It is recommended that this unit is assessed as a maximum of two assignments. The first assignment should assess learners' understanding of learning aim A and the second assignment should cover learning aims B and C.

The assignment for learning aim A could take the form of an academic paper or report examining the use of computational thinking skills and programming principles in a range of languages and contexts. However, a blog or some form of audio or visual evidence would also be acceptable and would allow learners to develop their creativity. The assignment brief should give enough scope for learners to explore how similar problems can be addressed using different languages and different programs. Learners will need to consider the impact that the implemented code has on the quality and appropriateness of the solution (i.e. program performance and whether it is fit for purpose).

The assignment for learning aims B and C should take the form of a practical project. Learners should plan, develop, test and review a software-based solution to meet the needs of a client. The scenario for the assignment should give enough scope to allow the learners to be able to consider different solutions and demonstrate a range of programming constructs. Learners will be expected to plan for and implement a range of testing methodologies to refine and optimise their solution. It is important that the context is realistic, and that learners have a 'client' for whom they are developing the system and whom they will work with throughout the project. The scope of the assignment should allow learners to show an understanding of writing computer software to meet specific, identified client requirements. To achieve a pass the program must function and broadly meet client requirements although some small, non-critical errors and inefficiencies may remain.

Getting started

This gives you a starting place for one way of delivering the unit, based around the recommended assessment approach in the specification.

Unit 4: Programming

Introduction

All areas of computing need people who are able to identify aspects of a problem and analyse the needs of users and clients to find a suitable solution. This unit gives learners solid foundation skills for deconstructing problems, planning and developing solutions, and applying the principles of computer programming to implement software that meets identified needs. These transferable skills will equip learners for further study or employment in the computing industry.

Learning aim A – Examine the computational thinking skills and principles of computer programming

Learners should be equipped with a range of skills and knowledge before starting the assignment – do not use the assignment as a vehicle to teach the content.

- You could begin by introducing the aim of the unit (i.e. the basic principles of computer programming and problem-solving) and how the concepts of computational thinking and programming constructs can be applied to practical problems. At first the problems should be more familiar to the learners using simple examples such as cooking a bowl of pasta, travelling from A to B or saving for a holiday. This should encourage learners to consider the wider issues such as what type of pasta, what quantity; what mode of transport, cost constraints; destination, length of holiday.
- Explain to learners that the four stages of computational thinking are decomposition, pattern recognition, pattern generalisation and abstraction, and algorithm design.
- Organise a variety of individual and group activities to allow learners to explore different software examples to analyse how computational thinking principles have been applied to produce solutions to problems. Database and spreadsheet examples will give learners a wide potential source of case studies.
- Ask learners to analyse a range of different software applications to examine how they fulfil the needs of the user and perform specific tasks and/or solve problems. You should give learners opportunities to compare programs with similar aims (eg different examples of a word processor) to see how common tools have been used and how the different instances vary in their ability to meet identified needs.
- Learners will need to understand the features, applications and implications of different high-level and low-level languages. You should give learners activities that focus on why a particular language may be used and the reasons for using it to create programs and other digital content.
- To develop learners' understanding of programming constructs, it may benefit them to start by looking at the sequencing of instructions. This will allow them to explore the importance of identifying the correct order in which tasks should be carried out. Allow time for learners to work on this individually, or in small groups, looking at given problems. You could also give them examples of solutions that do not always present tasks in the correct order, as this would require them to identify what is wrong with the proposed solution and consider its potential impact.

Learners should then explore programming constructs, and the logical and mathematical structures that can be used to describe computer processes. It may be beneficial to introduce these topics using pseudocode as a way of exploring the concepts without the additional demand of learning the syntax of a particular language. However, as learners start to understand the concepts covered in the topic, you should start to show them how these concepts can be implemented within different programming languages.

- You could give learners sample code and ask them to explain what the statements in the code do. Allowing learners to experiment with given code to explore the effects of any changes they make will help learners to develop an understanding of how a language works.
- You should give learners access to examples of software developed in different languages and which uses a variety of code bases. Learners should analyse how the example code has been implemented, and evaluate the implications of the techniques and processes used.
- When exploring the quality of software applications, as well as considering thecode base, ask learners to make use of testing. As part of this they should be reviewing and benchmarking tools to assess how well a solution meets its aims.
- Learners should explore real life vocational contexts, where possible. Engaging with local employers gives a perfect vehicle for this. Organise guest speakers from, or visits to, local employers, if you can. If not, you could replace or supplement these with high-quality, factual case studies.
- You should work with learners to develop their analytical skills by exploring different vocational contexts. This can be achieved by giving them opportunities to consider the requirements of a scenario and how the principles of computer programming could be implemented and the impact this would have.

Learning aim B – Design a software solution to meet client requirements

Ensure that learners are equipped with a range of skills and knowledge before starting the assignment rather than using the assignment as a vehicle to teach the content.

- To begin the learning aim, introduce the software development lifecycle and the application of its principles and process. Learners are not required to follow a particular lifecycle model, but are required to demonstrate practical project planning and management skills in a vocational scenario. Prepare and distribute handouts for the learners. Documentation for this unit is likely to be extensive and could include, but is not limited to, project proposals, technical specification documents, flowcharts, algorithm designs (pseudocode) and test plans.
- To develop strong vocational skills, incorporate project management techniques into lessons to ensure that learners can manage projects effectively. Include organising meetings with the client, recording outcomes from meetings and other forms of feedback, and adjusting plans and timescales for the project as appropriate.
- You should work with learners to ensure that they can display effective and appropriate communication skills. Check that all project documents and communication with clients uses appropriate style, tone and content.
- Before starting the assignment, you will need to give learners a practical introduction to a range of programming languages. Learners will need to know how to implement programming constructs in a number of languages, in order to appropriately select and justify a particular programming paradigm to meet the needs of their client.
- Learners will need to be aware of how to document their testing, review and refinement process. Allow time to show learners how to effectively perform and record ongoing testing, and seek and respond to feedback.

- Explain to learners that programs go through multiple stages of testing and refinement and therefore they should be aware of, and follow, appropriate procedures for recording changes and following versioning procedures.

Learning aim C – Develop a software solution to meet client requirements

Understanding of this learning aim should flow naturally from learning aim B and learners will need to be able to apply this understanding as part of a larger project. As with learning aim B, learners should be equipped with a range of project management and programming skills before starting the assignment. The assignment should not be used as a vehicle to teach the content.

- You should give learners opportunities to explore a range of testing methodologies to develop their understanding of how and why computer programs are tested in different ways at different stages during the development process.
- Learners should be able to select and apply appropriate testing methodologies so that solutions can be thoroughly tested and reviewed. Ensure that they are able to appropriately and thoroughly plan and document the testing process.
- To develop understanding of the testing and review process, you could give learners opportunities to use pre-existing programs, for which they could plan (and carry out) testing and review procedures.
- You should aim to develop learners' evaluative skills. Learners should be able to use the outcomes of testing and review to evaluate the quality of programmed solutions (and their own performance, as appropriate) against project requirements and client expectations.
- You should work with learners to ensure effective and appropriate presentation skills. You will need to emphasise the need for all project documents and communication with clients to use appropriate style, tone and content.
- The assignment should give learners a valid, vocational context. Try to organise a 'client' for learners to work with for the duration of the project, who will give them the operating requirements, set expectations and negotiate the timescales of the project.
- The client, where possible, should be a real-world client with whom the learner can engage. Although the project is simulated (ie it is not a live project), engaging with local employers to give a vocational setting would be invaluable. If you are not able to find real-world clients, ask another tutor or other adult to adopt the role of client. Other learners should not fulfil the role of client, although they could be test users. It is important that the client has a sound knowledge of the project and the related computing requirements
- Maintaining a diary or taking notes as they complete the various practical activities in the lessons relating to this learning aim will be of benefit to learners. They should also note the comments that their peers make when they give feedback.
- Ensure that learners understand how to fulfil the assessment criteria for the pass, merit and distinction grades.

Details of links to other BTEC units and qualifications, and to other relevant units/qualifications

Pearson BTEC Level 3 International in IT:

- *Unit 1: Information Technology Systems- Strategy, Management and Infrastructure*
- *Unit 6: Website Development*
- *Unit 7: Mobile Apps Development*
- *Unit 8: Computer Games Development*

Resources

In addition to the resources listed below, publishers are likely to produce Pearson-endorsed textbooks that support this unit of the BTEC Internationals in Information Technology. Check the Pearson website (http://qualifications.pearson.com/en/support/published-resources.html) for more information as titles achieve endorsement.

Websites

- www.python.org
  Python is an open-source programming language. The website has downloads for various operating systems and official documentation.
- www.w3schools.com/html/default.asp
  W3Schools - contains tutorials in web development languages, including HTML (with sections on HTML5), covering basic and more complex features.
- https://msdn.microsoft.com/en-us/library/2x7h1hfk.aspx
  Microsoft Developer Network – gives guidance for using Visual Basic, including language walkthroughs.
- www.microsoftvirtualacademy.com/en-us/training-courses/vb-fundamentals- for-absolute-beginners-8297
  Microsoft Virtual Academy – has a series of introductory tutorial videos on using Visual Basic.
- http://opensource.org
  The Open Source Initiative – gives help with open-source licensing and has links to various open-source projects.

Pearson is not responsible for the content of any external internet sites. It is essential for tutors to preview each website before using it in class so as to ensure that the URL is still accurate, relevant and appropriate. We suggest that tutors bookmark useful websites and consider enabling learners to access them through the school/college intranet.