Pearson BTEC

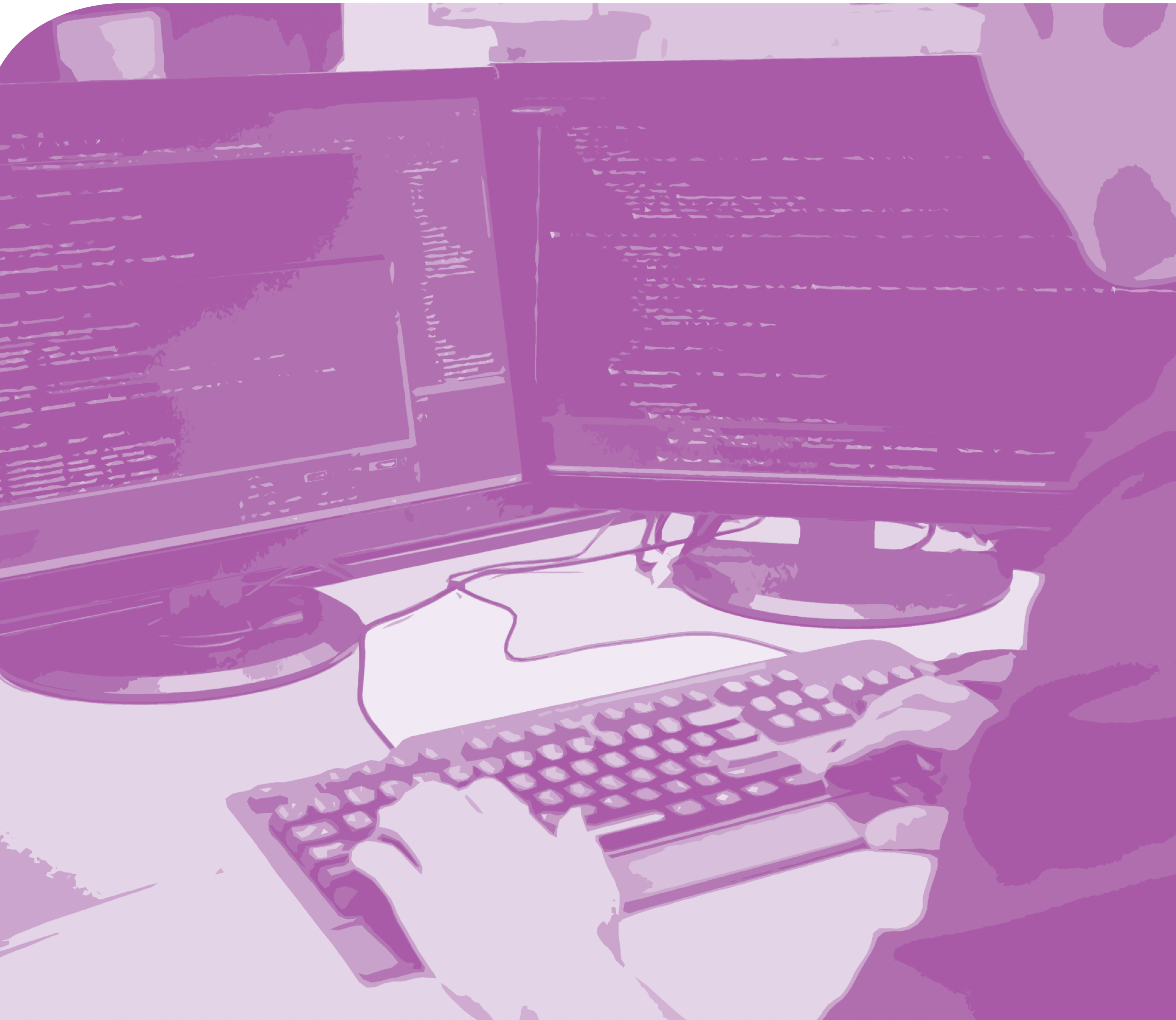
Pearson BTEC Uzbekistan Level 4 Qualifications in

Software Development

Unit 2: Software Analysis and Design

Teacher Resources

Issue 1



Edexcel, BTEC and LCCI qualifications

Edexcel, BTEC and LCCI qualifications are awarded by Pearson, the UK's largest awarding body offering academic and vocational qualifications that are globally recognised and benchmarked. For further information, please visit our qualifications website at qualifications.pearson.com. Alternatively, you can get in touch with us using the details on our contact us page at qualifications.pearson.com/contactus

About Pearson

Pearson is the world's leading learning company, with 35,000 employees in more than 70 countries working to help people of all ages to make measurable progress in their lives through learning. We put the learner at the centre of everything we do, because wherever learning flourishes, so do people. Find out more about how we can help you and your learners at qualifications.pearson.com

References to third party material made in this document are made in good faith. Pearson does not endorse, approve or accept responsibility for the content of materials, which may be subject to change, or any opinions expressed therein. (Material may include textbooks, journals, magazines and other publications and websites.)

All information in this document is correct at time of publication.

ISBN 978 1 446 96247 3

Contents

Introduction

This resource booklet is a companion to the BTEC Uzbekistan Level 4 Qualifications in Software Development. The specification tells you what must be taught and what must be assessed. This resource booklet gives you suggestions and ideas as to how you can do this.

This booklet gives you ideas for teaching and learning, including practical activities, realistic scenarios, ways of involving employers in delivery and of managing independent learning, and how to approach assessments. The booklet also shows you how the specification content might work in practice and inspires you to start thinking about different ways of delivering your qualification.

This resource booklet gives you:

- guidance on how to deliver the units in the qualification
- recommended resources to support the delivery of the units in the qualification
- schemes of work that show the topics, activities and assessments covered in all units across the qualification
- lesson plans with detailed guidance on how to deliver the lessons in the units.

The information in this resource booklet has been put together by teachers who have been close to the development of the qualifications and so understand the challenges of finding new and engaging ways to deliver BTEC qualifications.

The delivery guidance in this booklet gives you information on what you need to consider as you plan the delivery of the qualification. There is information on:

- the structure of your qualification
- how you can build the qualification for your learners
- suggestions for how you might make contact with appropriate employers
- information on other support and resources available.

We have given you unit-by-unit guidance. This includes suggestions on how to approach the learning aims and unit content, as well as ideas for interesting and varied activities. You will also find tips and ideas on how to plan for and deliver your assignments.

We have included a list of carefully selected resources for each unit. These resource lists offer suggestions for books, websites and videos that you can direct your learners to use and/or that you can use to complement delivery.

Unit 2: Software Analysis and Design

Delivery guidance

Approaching the unit

The purpose of this unit is for learners to investigate the principles of software analysis and design. They will then use these principles and develop their skills in order to analyse and design a solution to meet the needs of a client.

For learning aim A, start by discussing how to identify a client's business needs and the techniques that learners could use to gather required information from non-technical staff members. Make sure that learners are familiar with potential threats to projects and to the overall success of a system. Spend time discussing why projects might fail. Some real-life examples or a guest speaker would be useful here. Learners must investigate a client's software needs and produce a requirements specification according to the needs identified. To this end, ensure that learners understand how to produce the necessary requirements documentation.

For learning aim B, give learners as much practical experience as possible. Introduce the use of diagramming techniques early on and allow learners plenty of time to experiment with the tools available. In this learning aim, learners will design their software solutions to meet the requirements identified in learning aim A.

Learning aim C focuses on the testing of a software solution. Learners need to understand the different testing methods and understand when each method is used. They must also be able to develop a test plan for their proposed software design, including suitable test data.

Getting started

This gives you a starting place for one way of delivering the unit. It is based on the recommended assessment approach given in the specification.

Unit 2: Software Analysis and Design

Introduction

Software analysts investigate organisations to determine their needs for new software applications. They design solutions to meet an organisation's needs and they plan changes and updates to the solutions that they develop. Software analysts are key personnel in software development projects and they have good technical knowledge and problem-solving skills.

Learning aim A – Carry out a software requirements analysis

- Learners investigate the software requirements of a variety of different organisations. Give learners sample case studies and examples of good practice within industry. It would be beneficial to work with real employers.
- Learners carry out a document analysis and follow correct protocols when conducting this analysis (protocols such as confidentiality, company policy and security). Work with learners to ensure that they have effective communication skills.
- Use case studies showing failed projects as well as successful projects to allow learners to investigate what went wrong. If possible, invite a guest speaker to talk to the learners about successful and failed projects in order to emphasise the fact that projects can fail as well as succeed.
- Following their analysis, learners learn how to draft a requirements specification for a proposed new software application. Again, if possible, it would be useful for learners to have access to case studies or real-life examples. These could be provided by a guest speaker or by visits to a local business or organisation.

Unit 2: Software Analysis and Design
Learning aim B – Design a solution to an identified problem
• Learners learn how to design a new software application for a client in order to meet the client’s needs or improve their existing software applications. Give learners sample case studies and examples of good practice within industry, wherever possible. • Teach learners about designing user interfaces for different purposes. They must be introduced to methods of modelling, storing and processing the data within their software. Learners must be able to select diagramming techniques and use them accordingly, including flowcharts and data flow diagrams. • Instruct learners how to write pseudocode to demonstrate the programming constructs used within the software application. • Introduce learners to the different types of design documentation so that they can use these for a proposed software solution. Learners must be guided on how to make informed decisions on their design ideas, based on feedback, in order to make improvements to their designs. This can be done through presentation seminars where learners discuss the merits of specific designs and identify any areas that need improvement.
Learning aim C – Produce a test plan for an identified problem
• Teach learners how to design adequate testing processes to ensure that a software application is thoroughly tested. Learners should be introduced to testing methods and how to create test plans with suitable test data.

Details of links to other BTEC units and qualifications

This unit has links to other units in this qualification:

- Unit 1: Introduction to Programming
- Unit 3: Website Development.

Resources

Textbook

Pressman R. S. and Maxim B. R. – *Software Engineering: A Practitioner's Approach,* eighth edition, McGraw Hill Education (2014),
ISBN-13: 9780078022128

This is a comprehensive guide to software development. Digital and online copies are available.

Websites

www.tutorialspoint.com/software_engineering

This website features tutorials on software engineering.

www.w3computing.com/systemsanalysis

This website features a comprehensive W3 computing tutorial for systems analysis.

www.microtoolsinc.com/Howsrs.php

This website features a guide to making a good software requirements specification.

www.draw.io

This website features an online application for creating flowcharts.

Pearson is not responsible for the content of any external internet sites. It is essential for teachers 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 teachers bookmark useful websites and consider enabling learners to access them through the school/college intranet.

Scheme of work

Unit	Unit 2: Software Analysis and Design
Guided Learning Hours	60
Number of lessons	30
Duration of lessons	2 hours
Links to other units	• Unit 1: Introduction to Programming • Unit 3: Website Development

Key to learning opportunities			
AA	Assessment Activity	**RS**	Revision Session
GS	Guest Speaker	**V**	Visit
IS	Independent Study	**WE**	Work Experience

#	Topic	Lesson type	Suggested activities	Resources
1	A1 – Investigation techniques for software requirements analysis	IS	• **Teacher-led discussion:** to introduce the unit, initiate a discussion on the purpose of software systems and why they are important to organisations. • **Teacher presentation:** give an overview of what software analysis is and why it is necessary. Talk briefly about analysis methods. • **Individual activity:** learners consider case studies. • **Plenary activity:** summarise the lesson and use a question and answer activity to establish learners' understanding.	• Unit specification. • Dry-wipe board or flip chart for collating feedback. • Presentation on unit introduction. • Presentation on software analysis. • Software analysis case studies.
2	A1 – Investigation techniques for software requirements analysis	IS/GS	• **Teacher-led discussion:** present the ways of identifying an organisation's software needs. • **Guest speaker:** an IT professional from a local business talks to learners about the purpose of their business and its software needs. Learners ask pre-prepared questions. • **Small-group/paired activity:** learners document the software needs of the local business. • **Plenary activity:** summarise the lesson and use a question and answer activity to establish learners' understanding.	• Unit specification. • Dry-wipe board or flip chart for collating feedback. • Presentation on identifying software needs. • Presentation for guest speaker.

#	Topic	Lesson type	Suggested activities	Resources
3	A1 Investigation techniques	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Teacher presentation:** talk about the research and analysis required to provide a detailed and accurate description of an organisation's current software. • **Small-group/paired activity:** learners review a case study of a software development project in order to provide an analysis of current software needs. • **Plenary activity:** summarise the lesson and use a question and answer activity to establish learners' understanding.	• Unit specification. • Dry wipe board or flip chart to collate feedback. • Presentation on research and analysis of current software needs. • Case study. • Participatory survey methods for gathering information.

#	Topic	Lesson type	Suggested activities	Resources
4	A2 Threats to software success	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Teacher-led discussion/presentation:** discuss threats, including risks and constraints, to the success of a software development project. • **Small-group/paired activity:** give learners a software development plan and ask them to identify the risks to and constraints on its success. Learners then review case study and draft information-gathering plan. • **Plenary activity:** discuss how learners coped with the activities and use a question and answer activity to identify any gaps in understanding.	• Unit specification. • Flip chart or similar for learners to record discussions and ideas. • Learners’ work from previous lesson. • Presentation on threats to success of a software development. • Software development plan. • Case study.

#	Topic	Lesson type	Suggested activities	Resources
5	A2 Threats to software success	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Teacher-led discussion/presentation:** discuss how risks can be minimised or eliminated. • **Small-group/paired activity:** learners review the risks they identified in the previous lesson and suggest ways to minimise these risks. • **Plenary activity:** discuss how learners coped with the activity and use a question and answer activity to identify any gaps in understanding.	• Unit specification. • Flip chart or similar for learners to record discussions and ideas. • Learners' work from previous lesson.
6	A3 Requirements documentation	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Teacher presentation:** talk about the documentation that describes a new or updated system. • **Small-group/paired activity:** ask learners to draft software requirements documentation. • **Plenary activity:** confirm main learning points of the lesson and use a question and answer activity to establish learners' understanding.	• Unit specification. • Flip chart or similar for learners to record discussions and ideas. • Real-life case study from Lesson 4. • Case study from Lesson 4 • Documentation templates

#	Topic	Lesson type	Suggested activities	Resources
7	A3 Requirements documentation	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Teacher-led discussion:** explore how to complete the requirements documentation for a scenario about a real-life organisation. • **Small-group/paired activity:** learners complete the requirements documentation for the real-life scenario. • **Plenary activity:** discuss how learners coped with the activity and use a question and answer activity to identify any gaps in understanding.	• Unit specification. • Scenario for real-life organisation. • Flip chart or similar for learners to record discussions and ideas. • Documentation templates.

#	Topic	Lesson type	Suggested activities	Resources
8	B1 Flowcharts using standard symbols	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Teacher presentation:** explain what flowcharts are. • **Individual activity:** learners research the five main symbols used in flowcharts. • **Teacher-led discussion:** discuss what learners have found out and go through the standard flowchart symbols. • **Teacher presentation:** explain what flow charts are. • **Small-group/paired activity:** learners work in groups or pairs to develop a flowchart based upon a given scenario. • **Plenary activity:** confirm main learning points of lesson.	• Unit specification. • Presentation on flowcharts. • Scenario for flowchart activity. • Small dry wipe boards for paired activity. • Computers with Microsoft Visio® or Microsoft Word® installed. • Flowchart application: www.draw.io

#	Topic	Lesson type	Suggested activities	Resources
9	B1 Flowcharts using standard symbols	IS	- **Starter activity:** learners complete a quiz to recap what was covered in the previous session. - **Small-group/paired activity:** divide the group into pairs or small groups and give each group a different scenario that requires a flowchart to be developed. - **Plenary activity:** each group presents their solution to the class. - **Concluding activity:** confirm main learning points of lesson.	- Unit specification. - Quiz on flowcharts. - Dry-wipe board or flip chart to record discussions. - Computers with Microsoft Visio® or Microsoft Word® installed. - Scenarios for flowchart activity.

#	Topic	Lesson type	Suggested activities	Resources
10	B2 Structured English (pseudocode): • writing pseudocode • developing pseudocode.	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Individual activity:** learners independently research what pseudocode is and give some examples. Discuss the results. • **Teacher presentation:** explain what pseudocode is and link this to the flowchart scenarios given in previous lessons. • **Small-group/paired activity:** give the whole group a basic scenario and ask learners to work in pairs or small groups to develop a solution to the scenario using pseudocode. • **Plenary activity:** confirm main points of lesson and use question and answer activity to establish learners' understanding.	• Unit specification. • Dry-wipe board or flip chart to collate feedback. • Presentation. • Flowchart scenarios from previous lessons. • Scenario for pseudocode activity. • Small dry wipe boards for paired activity. • Computers with Microsoft Visio® or Microsoft Word® installed.

#	Topic	Lesson type	Suggested activities	Resources
11	B2 Structured English (pseudocode): • writing pseudocode • developing pseudocode.	IS	• **Starter activity:** learners complete a quiz to recap the previous lesson. • **Individual activity:** learners identify errors in example pseudocode. • **Learner presentation and teacher-led discussion:** learners present their findings to the rest of the group and discuss. • **Plenary activity:** confirm main points of lesson and use question and answer activity to check learners' understanding.	• Unit specification. • Dry-wipe board or flip chart for collating record discussions. • Quiz on pseudocode. • Examples of pseudocode (including some errors for learners to identify).
12	B3 Data flow diagrams	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Teacher presentation:** give an overview of data flow diagrams. • **Small-group/paired activity:** learners draft a data flow diagram for software proposal. • **Plenary activity:** discuss how learners coped with the activity.	• Unit specification. • Presentation on data flow diagrams. • Software proposal from Lesson 7. • Data flow diagrams.

#	Topic	Lesson type	Suggested activities	Resources
13	B3 Data flow diagrams	IS	- **Starter activity:** learners complete a quiz to recap the previous lesson. - **Individual activity:** learners identify errors in example data flow diagrams. - **Learner presentation and teacher-led discussion:** learners present their findings to the rest of the group and discuss. - **Plenary activity:** confirm main points of lesson and use question and answer to check learners' understanding.	- Unit specification. - Quiz on data flow diagrams. - Dry-wipe board or flip chart for collating feedback. - Examples of data flow diagrams (including some errors for learners to identify).

#	Topic	Lesson type	Suggested activities	Resources
14	B4 Input and output requirements	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Teacher presentation:** give an overview of the appropriate and detailed design documentation needed for the input and output of the proposed system. • **Small-group/paired activity:** learners recommend input and output requirements from the requirements specification that they developed in Lesson 7. • **Plenary activity:** discuss how learners coped with the activity and use question and answer activity to identify any gaps in understanding.	• Unit specification. • Presentation on input and output requirements. • Scenario and learners' requirements documentation from Lesson 7. • Dry-wipe board or flip chart for collating feedback.

#	Topic	Lesson type	Suggested activities	Resources
15	B4 Input and output requirements	IS	• **Starter activity:** learners complete a quiz to recap the previous lesson. • **Individual activity:** learners identify errors in provided examples of design documentation. • **Learner presentation and teacher-led discussion:** learners present their findings to the rest of the group and discuss. • **Plenary activity:** confirm main points of lesson and use question and answer to check learners' understanding.	• Unit specification. • Quiz on input and output requirements. • Dry-wipe board or flip chart for collating feedback • Examples of design documentation (including some errors for learners to identify)
16	C1 Testing methodologies	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Teacher presentation:** explain why software testing is important and ask for learners' contributions. • **Small-group/paired activity:** learners research the reasons why software testing is important. • **Plenary activity:** confirm main points of lesson and use question and answer to check learners' understanding.	• Unit specification. • Presentation on software testing. • Small dry wipe boards for small-group/paired activity. • Computers with internet access.

#	Topic	Lesson type	Suggested activities	Resources
17	C1 Testing methodologies: • functional testing.	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Teacher presentation:** explain functional testing and ask for learners' contributions. • **Small-group/paired activity:** learners research functional testing and present evidence on each test carried out. • **Plenary activity:** confirm main points of lesson and use question and answer to check learners' understanding.	• Unit specification. • Presentation on functional testing. • Small dry-wipe boards for small-group/paired activity. • Computers with internet access.
18	C1 Testing methodologies: • non-functional testing.	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Teacher presentation:** explain non-functional testing and ask for learners' contributions. Discuss the tests carried out in non-functional testing and identify real-life examples of this testing method. • **Small-group/paired activity:** learners research non-functional testing and present evidence on each test carried out. • **Plenary activity:** confirm main points of lesson and use question and answer to check learners' understanding.	• Unit specification. • Presentation on non-functional testing. • Small dry-wipe boards for small-group/paired activity. • Computers with internet access.

#	Topic	Lesson type	Suggested activities	Resources
19	C1 Testing methodologies: • what to test • choice of test data • expected test results from a range of testing methods.	IS	• **Starter activity:** recap previous lesson and introduce topic. • **Teacher presentation:** show how a test plan is created and what test data must be used. • **Individual activity:** learners test a program on their computer using a test plan. • **Small-group/paired activity:** learners work in pairs or small groups to produce a 5–10-minute presentation of their test results. • **Plenary activity:** confirm main points of lesson and use question and answer to check learners' understanding.	• Unit specification. • Presentation on test plans. • Program for learners to test. • Test plan template. • Computers with Microsoft Visio® or Microsoft Word® and Microsoft PowerPoint® or other suitable presentation software.
20	Learning aim A Practice assessment	AA	• **Starter activity:** recap previous lesson and introduce topic. • **Practice assessment activity:** give learners a practice assessment scenario. • **Teacher-led discussion:** work through the answers to the practice assessment with learners. • **Plenary activity:** confirm main points of lesson and use question and answer to check learners' understanding.	• Unit specification. • Practice assessment activity. • Dry-wipe board or flip chart for collating feedback. • Computers with Microsoft Visio® or Microsoft Word® and Microsoft PowerPoint® or other suitable presentation software.

#	Topic	Lesson type	Suggested activities	Resources
21	Learning aim B Practice assessment	AA	• **Starter activity:** recap previous lesson and introduce topic. • **Practice assessment activity:** give learners a practice assessment scenario. • **Teacher-led discussion:** work through the answers to the practice assessment with learners. • **Plenary activity:** confirm main points of lesson and use question and answer to check learners' understanding.	• Unit specification. • Practice assessment activity. • Dry-wipe board or flip chart for collating feedback. • Computers with Microsoft Visio® or Microsoft Word® and Microsoft PowerPoint® or other suitable presentation software.
22	Learning aim C Practice assessment	AA	• **Starter activity:** recap previous lesson and introduce topic. • **Practice assessment activity:** give learners a practice assessment scenario. • **Teacher-led discussion:** work through the answers to the practice assessment with learners. • **Plenary activity:** confirm main points of lesson and use question and answer to check learners' understanding.	• Unit specification. • Practice assessment activity. • Dry-wipe board or flip chart for collating feedback. • Computers with Microsoft Visio® or Microsoft Word® and Microsoft PowerPoint® or other suitable presentation software.

#	Topic	Lesson type	Suggested activities	Resources
23	Learning aims A, B and C Final assessment	AA	• **Starter activity:** recap previous lessons. • **Teacher-led discussion:** introduce the final assessment using the Assessment Workbook. • **Final assessment activity:** using the Assessment Workbook, learners complete all tasks.	• Unit specification. • Assessment Workbook. • Computers with internet access. • Test plan template.
24–29	Learning aims A, B and C Final assessment	AA	• **Final assessment activity:** using the Assessment Workbook, learners complete all tasks.	• Computers with internet access. • Assessment Workbook. • Test plan template.
30	Learning aims A, B and C Final assessment feedback		• **Starter activity:** recap previous lessons. • **Teacher-led discussion:** work through the answers to the assessment with learners and identify signs of good work.	• Unit specification. • Dry-wipe board or flip chart for collating feedback. • Computers with internet access. • Assessment Workbook.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	1 (2 hours)

Lesson objectives	At the end of the lesson, learners will understand why software analysis is needed.

Resources checklist	• Unit specification. • Dry-wipe board or flip chart for collating feedback. • Presentation on unit introduction. • Presentation on software analysis. • Software analysis case studies.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (30 minutes)	• **Teacher presentation:** introduction to unit. • **Teacher-led discussion:** explore what software systems are and why they are important. What are learners' experiences of software analysis and design?
Main activity (1 hour 20 minutes)	• **Teacher presentation**: overview of what software analysis is and why it is necessary. • **Teacher-led discussion:** question learners with regards to the key questions to be asked in software analysis and the methods of analysis. • **Individual activity:** learners use case studies to understand why software analysis is required.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	2 (2 hours)

Lesson objectives	At the end of the lesson, learners will be able to identify the software needs of an organisation.

Resources checklist	• Unit specification. • Dry-wipe board or flip chart for collating feedback. • Presentation on identifying software needs. • Presentation for guest speaker.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (30 minutes)	• **Teacher-led discussion:** give a presentation explaining how to identify the software needs of an organisation. • **Small-group/paired activity:** learners prepare questions for guest speaker.
Main activity (1 hour 20 minutes)	• **Guest speaker presentation:** guest speaker gives an overview of what their business does and the software needs of the business. Learners ask the guest speaker their pre-prepared questions about the speaker's business. • **Small-group/paired activity:** learners document the software needs of the speaker's business.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding of the topic.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	3 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand how to carry out research and analysis of current software needs • be able to create research and analysis documentation.

Resources checklist	• Unit specification. • Dry-wipe board or flip chart for collating feedback. • Presentation on research and analysis of current software needs. • Case study. • Participatory survey methods for gathering information.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson and hold a question and answer activity on the previous lesson. Introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher presentation:** give a presentation on the research and analysis of current software needs. The presentation covers information-gathering methods (such as questionnaires, interviews and observations) and talks about protocols for investigations, such as confidentiality and security of sensitive information. • **Individual activity:** learners consider the use of questionnaires, observation and structured interviews. They also consider potential defects in information gathered, such as individuals not always telling the truth or individuals forgetting about tasks they do not do regularly, meaning that information would be incomplete. They should understand the care that is needed when designing questions (such as whether questions should be open or closed and the ramifications of this choice on the type of responses) and the potential for poor responses to questionnaires that are sent out. • **Small-group/paired activity:** learners review a case study of a software development project. They draft a plan for gathering information from an organisation, ensuring that any relevant protocols are followed.
Plenary activity (10 minutes)	• **Teacher-led discussion:** summarise the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	4 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the threats to software success • be able to identify threats to software developments.

Resources checklist	• Unit specification. • Flip chart or similar for learners to record discussions and ideas. • Learners’ work from previous lesson. • Presentation on threats to success of a software development. • Software development plan. • Real-life case study.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson and hold question and answer activity on previous lesson topics. Introduce aims and objectives of this lesson.
Main activity (1 hour 30 minutes)	• **Teacher presentation:** discuss threats to the success of a software development, including risks, constraints, provision of appropriate resources and change management. Include real-life examples where possible. • **Small-group/paired activity:** give learners a software development plan and ask them to identify risks to and constraints on its success. Learners should give feedback to the class on the risks they have identified and then quantify these risks. • **Small-group/paired activity:** learners review a real-life case study for a software development project. They should draft a plan for gathering information from an organisation, ensuring that relevant protocols are followed.
Plenary activity (10 minutes)	• **Teacher-led discussion:** discuss how learners coped with the activities. Summarise the main learning points of the lesson and use a question and answer activity to identify any gaps in understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	5 (2 hours)

Lesson objectives	At the end of the lesson, learners will understand how to minimise the threats to software success.

Resources checklist	• Unit specification. • Flip chart or similar for learners to record discussions and ideas. • Software development plan from previous lesson. • Learners' work from previous lesson.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous session and hold a question and answer activity on the previous lesson. Introduce the aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher-led discussion/presentation:** discuss how to minimise threats to the success of a software development, including risks, constraints, provision of appropriate resources and change management. Include real-life examples where possible. • **Small-group/paired activity:** give learners the software development plan from the previous lesson and ask them to review their identified risks. Learners suggest ways to minimise the risks that they identified.
Plenary activity (10 minutes)	• **Teacher-led discussion:** discuss how learners coped with the activity. Confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	6 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the requirements documentation needed • be able to draft software requirements documentation.

Resources checklist	• Unit specification • Dry-wipe board or flip chart for collating feedback • Real-life case study from Lesson 4. • Documentation templates.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson with a question and answer activity and introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher-led discussion:** discuss the documentation that describes a new or updated system, ensuring that all content listed in the unit specification is covered. • **Small-group/paired activity:** ask learners to draft a software requirements specification for the real-life organisation in the case study from Lesson 4.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	7 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand how to complete the requirements documentation • be able to draft the documentation needed for the software requirements specification.

Resources checklist	• Unit specification. • Scenario for real-life organisation. • Dry-wipe board or flip chart for collating feedback. • Documentation templates.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous session using question and answer activity. Introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher-led demonstration/discussion:** discuss how to complete the software requirements documentation for a real-life organisation. Use the documentation templates to demonstrate how this is done. • **Small-group/paired activity:** learners complete the requirements documentation for the real-life organisation.
Plenary activity (10 minutes)	• **Teacher-led discussion:** discuss how learners coped with the activity and use a question and answer activity to identify any gaps in understanding. Confirm the main learning points of the lesson.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	8 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand why flowcharts are needed in software design • be able to create flowcharts using standard symbols.

Resources checklist	• Unit specification. • Presentation on flowcharts. • Scenario for flowchart activity. • Small dry-wipe boards for paired activity. • Computers with Microsoft Visio® or Microsoft Word® installed. • Flowchart application: www.draw.io.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson with question and answer activity and introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher presentation:** explain what flowcharts are and how the flowchart symbols should be applied. Go on to explain how software applications such as Microsoft Word® and Microsoft Visio® can be used to create a flowchart. Learners can make use of free charting software. • **Individual activity:** learners research the five main symbols used in flowcharts. • **Teacher-led discussion:** discuss what learners have found and go through the list of symbols in the specification appendix. • **Small-group/paired activity:** give learners a scenario that requires them to create a flowchart to provide a solution for software problems.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	9 (2 hours)

Lesson objectives	At the end of the lesson, learners will be able to: • identify the standard symbols used in flowcharts • create more advanced flowcharts using standard symbols.

Resources checklist	• Unit specification. • Quiz on flowcharts. • Dry-wipe board or flip chart to record discussions. • Computers with Microsoft Visio® or Microsoft Word® installed. • Scenarios for flowchart activity.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Starter activity (30 minutes)	• **Teacher-led discussion:** introduce aims and objectives for this lesson. • **Individual activity:** give learners a quiz to recap what was covered in the previous session, with questions designed to check that learners remember the symbols used in flow charts. This can be self- or peer marked.
Main activity (1 hour 20 minutes)	• **Small-group/paired activity:** divide the group into pairs or small groups and give each group a different scenario that requires a flowchart to be developed. The flowchart must be suitably complex and created using software packages such as Microsoft Visio® or Microsoft Word®. • **Plenary activity:** each pair or small group presents their solution to the class. Facilitate a group discussion on whether each solution derived is correct and appropriate.
Concluding activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points identified in the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	10 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand why pseudocode is needed in software design • be able to write pseudocode.

Resources checklist	• Unit specification. • Dry-wipe board or flip chart for collating feedback. • Presentation. • Flowchart scenarios from previous lessons. • Scenario for pseudocode activity. • Small dry-wipe boards for paired activity. • Computers with Microsoft Visio® or Microsoft Word® installed.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Starter activity (20 minutes)	● **Teacher-led discussion:** recap previous lesson with a question and answer activity. Introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	● **Individual activity:** learners independently research what pseudocode is and give some examples. ● **Teacher-led discussion:** discuss what learners have found out about pseudocode. ● **Teacher presentation:** explain what pseudocode is and link this to the flowchart-related scenarios given in previous lessons. When explaining how pseudocode is developed, emphasise the importance of correct sequencing to determine the flow of a program. ● **Small-group/paired activity:** give the whole group a basic scenario and ask learners to work in pairs or small groups to develop a solution to the scenario, using pseudocode.
Plenary activity (10 minutes)	● **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	11 (2 hours)

Lesson objectives	At the end of the lesson, learners will be able to: • identify the standard pseudocode notation used within software design • write pseudocode using standard notation.

Resources checklist	• Unit specification. • Dry wipe board or flip chart to record discussions. • Quiz on pseudocode. • Examples of pseudocode (including some errors for learners to identify).
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (30 minutes)	• **Teacher-led discussion:** introduce aims and objectives for this lesson. • **Quiz:** give learners a quiz to recap what was covered in the previous lesson with questions designed to check that learners remember the main points about pseudocode. This can be self- or peer marked.
Main activity (1 hour 20 minutes)	• **Individual activity:** give learners examples of pseudocode that include errors. Learners should work independently to identify the errors and present what they find to the rest of the group. • **Learner presentation and teacher-led discussion:** learners present their findings to the whole group. Facilitate a group discussion about whether each finding is correct.

Activities	Teaching notes
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	12 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand why data flow diagrams are needed in software design • be able to create data flow diagrams.

Resources checklist	• Unit specification. • Presentation on data flow diagrams. • Data flow diagrams. • Small dry-wipe boards for paired activity/ basic stationery for notetaking.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson with question and answer activity. Introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher presentation:** give an overview of the detailed design documentation showing the data requirements of and use within a proposed software application. Cover data modelling, including data flow diagrams (DFDs). • **Small-group/paired activity:** learners should investigate the software proposal from Lesson 7 for its data requirements and draft a DFD.
Plenary activity (10 minutes)	• **Teacher-led discussion:** discuss how learners coped with the activity. Confirm the main learning points of the lesson and use a question and answer activity to establish understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	13 (2 hours)

Lesson objectives	At the end of the lesson, learners will be able to: • identify the standard data flow diagram symbols used in software design • create data flow diagrams.

Resources checklist	• Unit specification. • Quiz on data flow diagrams. • Dry-wipe board or flip chart for collating feedback. • Examples of data flow diagrams (including some errors for learners to identify).
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Starter activity (30 minutes)	• **Quiz:** give learners a quiz to recap what was covered in the previous session, with questions designed to check that learners remember the use of data flow diagrams. This can be self- or peer-marked. • **Teacher-led discussion:** introduce aims and objectives for this lesson.
Main activity (1 hour 20 minutes)	• **Individual activity:** give learners examples of data flow diagrams that include errors. Learners work independently to identify the errors. • **Learner presentation:** learners present their findings to the rest of the group. Facilitate a group discussion about whether each finding is correct.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	14 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the documentation needed in software design • be able to use the documentation required for software design.

Resources checklist	• Unit specification. • Presentation on input and output requirements. • Scenario and learners' requirements documentation from Lesson 7. • Dry-wipe board or flip chart for collating feedback.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson with question and answer activity and introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher presentation:** give an overview of the appropriate and detailed design documentation needed for the input and output of the proposed system, including visuals, data entry forms and report forms. Introduce learners to some of the tools for design mock-ups. • **Small-group/paired activity:** learners recommend input and output requirements based on the requirements specification that they developed in Lesson 7.
Concluding activity (10 minutes)	• **Plenary activity:** discuss how learners coped with the activity. Confirm the main learning points of the lesson and use a question and answer activity to establish understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	15 (2 hours)

Lesson objectives	At the end of the lesson, learners will be able to: • identify what documentation is needed to design software • complete design documentation used in software design.

Resources checklist	• Unit specification. • Quiz on input and output requirements. • Dry-wipe board or flip chart for collating feedback. • Examples of design documentation (including some errors for learners to identify).
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Starter activity (30 minutes)	- **Quiz:** give learners a quiz to recap what was covered in the previous session. The questions should cover input and output requirements and include some example case studies. This can be self- or peer-marked. - **Teacher-led discussion:** introduce the aims and objectives for this lesson.
Main activity (1 hour 20 minutes)	- **Individual activity:** give learners examples of design documentation that include errors. Learners work independently to identify the errors and present what they find to the rest of the group. - **Learner presentation and teacher-led discussion:** learners present their findings to the rest of the group. Facilitate a group discussion on whether each finding is correct.
Plenary activity (10 minutes)	- **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	16 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the importance of testing in software design • be able to identify reasons for testing in software design.

Resources checklist	• Unit specification. • Presentation on software testing. • Small dry-wipe boards for small-group/paired activity. • Computers with internet access.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson using a question and answer activity. Introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher presentation:** explain why software testing is important and ask for learners' contributions. Discuss the consequences of programs or applications that are not robustly tested and cite real-life examples where possible, such as the Patriot Missile Defence System. • **Small-group/paired activity:** learners research the reasons why software testing is important.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	17 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand functional testing in software design • be able to carry out functional testing in software design.

Resources checklist	• Unit specification. • Presentation on functional testing. • Small dry-wipe boards for small-group/paired activity. • Computers with internet access.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson with a question and answer activity. Introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher presentation:** explain functional testing and ask for learners' contributions. Discuss the tests carried out within functional testing and identify real life examples of this testing method being carried out. • **Small-group/paired activity:** learners research functional testing and present evidence on each test carried out.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	18 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand non-functional testing in software design • be able to carry out non-functional testing in software design.

Resources checklist	• Unit specification. • Presentation on non-functional testing. • Small dry-wipe boards for small-group/paired activity. • Computers with internet access.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson with a question and answer activity. Introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher presentation:** explain non-functional testing and ask for learners' contributions. Discuss the tests carried out within non-functional testing and identify real life examples of this testing method. • **Small-group/paired activity:** learners research non-functional testing and present evidence on each test carried out.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	19 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the need for test plans in software design • be able to carry out a testing strategy in software design.

Resources checklist	• Unit specification. • Presentation on test plans. • Program for learners to test. • Test plan template. • Computers with Microsoft Visio® or Microsoft Word® and Microsoft PowerPoint® or any other suitable presentation software.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson using a question and answer activity and introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher presentation:** show learners a test plan and how to create it. Demonstrate how normal, abnormal and extreme test data should be applied to ensure that the program is thoroughly tested. • **Individual task:** learners test a program on their computer. Give them a test data template to work with and ask them to complete 15 tests designed to check for functionality, stability and usability using different choices of test data. Explain what is meant by 'expected test results'. • **Small-group/paired activity:** learners work in pairs or small groups to produce a 5–10-minute presentation of their test results. Learners can use their testing templates and refine them to produce results.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	20 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand how learning aim A will be assessed • be able to carry out the work required for learning aim A.

Resources checklist	• Unit specification. • Practice assessment activity. • Dry-wipe board or flip chart for collating feedback. • Computers with Microsoft Visio® or Microsoft Word® and Microsoft PowerPoint® or any other suitable presentation software.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson with question and answer activity. Introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Practice assessment activity:** give learners a practice assessment scenario. Under controlled conditions, learners produce requirements documentation based on the scenario. • **Teacher-led discussion:** work through the answers to the practice assessment with learners.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	21 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand how learning aim B will be assessed • be able to carry out the work required for learning aim B.

Resources checklist	• Unit specification. • Practice assessment activity. • Dry-wipe board or flip chart for collating feedback. • Computers with Microsoft Visio® or Microsoft Word® and Microsoft PowerPoint® or any other suitable presentation software.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson with a question and answer activity. Introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Practice assessment activity:** give learners a practice assessment scenario. Under controlled conditions, learners develop a pseudocode and flowchart solution based on the scenario. • **Teacher-led discussion:** work through the answers to the practice assessment with learners.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners’ understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	22 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand how learning aim C will be assessed • be able to carry out the work required for learning aim C.

Resources checklist	• Unit specification. • Practice assessment activity. • Dry-wipe board or flip chart for collating feedback. • Computers with Microsoft Visio® or Microsoft Word® and Microsoft PowerPoint® or any other suitable presentation software.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson with question and answer activity. Introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Practice assessment activity:** give learners a practice assessment scenario. Under controlled conditions, learners develop a testing strategy and plan a solution based on the scenario. • **Teacher-led discussion:** work through the answers to the practice assessment with learners.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	23 (2 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the assessment requirements for the unit • be able to carry out the work required for the assessment of learning aims A, B and C.

Resources checklist	• Unit specification. • Assessment Workbook. • Computers with internet access. • Test plan template.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (30 minutes)	• **Teacher-led discussion:** recap previous lesson with a question and answer activity. Introduce the aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher-led discussion:** introduce the final assessment (see authorised assessment briefs), detailing the main assessment criteria, deadline and submission criteria. • **Final assessment activity:** using the Assessment Workbook, learners start work on the tasks.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	24–29 (12 hours)

Lesson objectives	At the end of the lesson, learners will have completed the work required for the assessment of learning aims A, B and C.

Resources checklist	• Assessment Workbook. • Computers with internet access. • Test plan template.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Main activity (2 hours)	• **Final assessment activity:** using the Assessment Workbook, learners continue to work on the tasks.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 2: Software Analysis and Design
Lesson number	30 (2 hours)

Lesson objectives	At the end of the lesson, learners will know how well they have completed the assessment.

Resources checklist	• Unit specification. • Dry-wipe board or flip chart for collating feedback. • Assessment Workbook. • Learners' marked work
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Starter activity (20 minutes)	• **Teacher-led discussion:** recap previous lesson with question and answer activity. Introduce aims and objectives for this lesson.
Main activity (1 hour 30 minutes)	• **Teacher-led discussion:** work through the answers to the assessment with learners and identify signs of good work.
Plenary activity (10 minutes)	• **Teacher-led discussion:** confirm the main learning points of the lesson and use a question and answer activity to establish learners' understanding.

For information about Pearson Qualifications, including Pearson Edexcel, BTEC and LCCI qualifications visit qualifications.pearson.com

Edexcel and BTEC are registered trademarks of Pearson Education Limited

Pearson Education Limited. Registered in England and Wales No. 872828
Registered Office: 80 Strand, London WC2R 0RL.

VAT Reg No GB 278 537121

Pearson
BTEC

ISBN 978-1-4469-6247-3