Pearson BTEC

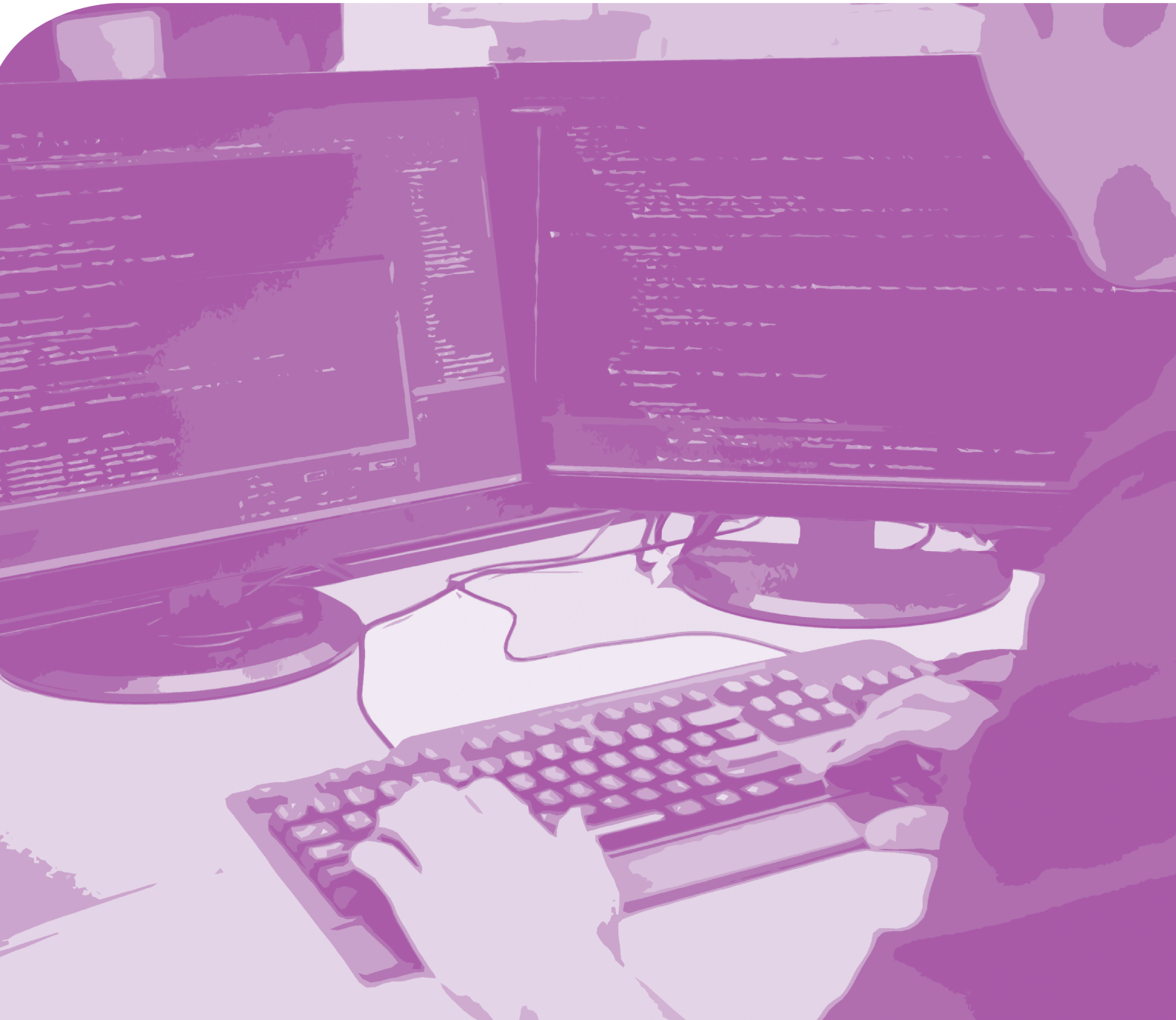
Pearson BTEC Uzbekistan Level 4 Qualifications in

Software Development

Unit 4: Programming Using Different Coding Paradigms

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 0 9978 8303 9

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 unit
- recommended resources to support the delivery of the unit
- a scheme of work that show the topics, activities and assessments covered in the unit
- lesson plans with detailed guidance on how to deliver the lessons in the unit

The information in this resource booklet has been put together by teachers who have been close to the development of the qualification 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 unit. 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 assessments.

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

Unit 4: Programming Using Different Coding Paradigms

Delivery guidance

Approaching the unit

The purpose of this unit is for learners to understand the key concepts of event-driven and object-oriented programming and to develop skills in these languages.

In this unit, learners are taught about the key features of event-driven programming and apply this in learning how to develop and test a program. They will then learn about the key features of object-oriented programming and apply this in learning how to develop and test a program using that coding paradigm. Learners will also consider how these two coding paradigms differ from each other, and from the procedural paradigm that they learned in Unit 1.

Delivering the learning aims

For Learning aim A, you could start by discussing event triggers, handlers and controls. Make sure that learners are familiar with how these are applied within an integrated development environment (IDE). You could spend time demonstrating use of the IDE and the features that learners need to be aware of.

Learners must create, test and refine a computer program in an event-driven language to meet client requirements. To this end, ensure that learners understand how to produce applications in the selected language and how to fully test them for functionality. Give learners as much practical experience as possible.

For Learning aim B, give learners as much practical experience as possible. Introduce the use of syntax elements early and allow learners time to experiment with the IDE tools available.

Learners must create, test and refine a computer program in an object-oriented language to meet client requirements. To this end, ensure that learners understand how to produce applications in the selected language and how to fully test them for functionality. Give learners as much practical experience as possible.

In Learning aim C the focus is on the key features of event-driven, object-oriented and procedural programming languages. Learners need to understand the differences between each language and reasons to use one over another. They must refer to their notes made in Unit 1 to help with fully understanding the key features of procedural programming.

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 4: Programming Using Different Coding Paradigms
Introduction Software developers use different programming languages to develop new software applications. They develop solutions to meet and organisation's needs, and plan changes and updates to the solutions they develop. Software developers are key personnel in software development projects and they need to have good problem-solving skills and technical knowledge of different programming languages.
Learning aim A – Develop and test a computer program in an event-driven language
This topic will give learners the tools they need to create a software program in an event-driven language. Give learners plenty of tuition in the chosen language to address each part of the unit specification. • Start by presenting learners with the theory behind the way in which data should be handled within a program. • When you are confident that learners understand the key concepts, give them several practical exercises to practise these concepts. For example, learners could be given a workbook to work through at their own pace as the topic progresses. This workbook should give them coding snippets to show how the concepts are applied to problems. This will also give you an opportunity to monitor learners' progress. • As learners hone their programming skills and work through various tasks, it would be useful to bring learners together periodically to recap the skills that they have mastered. Also, when an exercise proves challenging for some learners, you could work through model answers on an interactive whiteboard, with the assistance of other learners in the class. This will also give you a chance to ask learners direct questions to check their understanding. You could ask stronger programmers in the class to formally buddy/work with weaker learners.

Unit 4: Programming Using Different Coding Paradigms

- Once all learners have completed their workbooks, you should set them a mock assessment. The mock assessment must reflect the external assessment that they will complete, so it should supply an appropriate problem for which learners can create solutions. This task should be completed independently and individually under controlled conditions that replicate the live external assessment**.**

Learning aim B – Develop and test a computer program in an object-oriented language

This topic will give learners the tools they need to create a software program in an object-oriented language. Give learners plenty of tuition in the chosen language to address each part of the unit specification.

- Start by presenting learners with the theory behind the way in which data should be handled within a program.
- When you are confident that learners understand the key concepts, give them several practical exercises to practise these concepts. For example, learners could be given a workbook to work through at their own pace as the topic progresses. This workbook should give them coding snippets to show how the concepts are applied to problems. This will also give you an opportunity to monitor learners' progress.
- As learners hone their programming skills and work through various tasks, it would be useful to bring learners together periodically to recap the skills that they have mastered. Also, when an exercise proves challenging for some learners, you could work through model answers on an interactive whiteboard, with the assistance of other learners in the class. This will also give you a chance to ask learners direct questions to check their understanding. You could ask stronger programmers in the class to work with weaker learners.
- Once all learners have completed their workbooks, you should set them a mock assessment. The mock assessment must reflect the external assessment that they will complete, so it should supply an appropriate problem for which learners can create solutions. This task should be completed independently and individually under controlled conditions that replicate the final assessment.

Unit 4: Programming Using Different Coding Paradigms
Learning aim C – Understand the key features of different programming languages
By the end of this topic, learners will need to understand the features, applications and implications of three different languages: event-driven, object-oriented and procedural. • When delivering this topic, you may wish to start with activities that focus on why a language may be used and the concepts and principles behind using it to create programs and other digital content. • Ensure learners are aware that procedural programming was covered in Unit 1 so they can refer to their notes from that unit. • You could give learners follow-up activities, such as asking them to compare how a given piece of code in one programming language might be translated into another and the subsequent implications of implementing the code in a different language.

Details of links to other BTEC units and qualifications

This unit links to:

- Unit 1: Introduction to Programming
- Unit 2: Software Analysis and Design

Resources

Websites

www.cplusplus.com/doc/tutorial/

Clear explanations on the basics of C++, program structure, compound data types, as well as other features useful for programming in C++.

www.youtube.com/watch?v=tvC1WCdV1XU&list=PLAE85DE8440AA6B83

The first of 73 video tutorials on how to use C++. It is very useful to anyone wishing to learn C++ programming but who has little experience of using it.

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 4: Programming Using Different Coding Paradigms
Guided Learning Hours	150
Number of lessons	50
Duration of lessons	3 hours
Links to other units	• Unit 1: Introduction to Programming • Unit 2: Software Analysis and Design

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	C1 Key features of event-driven programming	IS	• **Teacher-led discussion:** To introduce the unit, initiate a discussion on event-driven programming, its characteristics and uses. • **Teacher presentation:** Give an overview of event-driven programming and introduce the key features. • **Individual activity:** Learners use case studies to understand why event-driven languages are required. • **Plenary:** Summarise the lesson and use oral questioning to confirm learners' understanding.	Dry wipe board or flip chart to collate feedback Teacher presentations and notes Case studies
2	A6 Using an integrated development environment (IDE)	IS	• **Teacher-led discussion:** To introduce the use of IDE in the development of event-driven programming. • **Teacher presentation:** Give an overview of an event-driven IDE, its features and benefits. • **Individual activity:** Learners work on practice tasks that require them to develop simple programs using the IDE. • **Plenary:** Summarise the lesson and use oral questioning to confirm learners' understanding.	Dry wipe board or flip chart to collate feedback Teacher presentations and notes Computers with IDE installed Practice task activity sheets

#	Topic	Lesson type	Suggested activities	Resources
3	A1 Event triggers	IS	• **Teacher-led discussion:** Discuss what event triggers are. • **Paired/small-group activity:** Learners should research and document the different event triggers and then create code that uses event triggers. • **Plenary:** Summarise the lesson and use oral questioning to confirm learners' understanding.	Dry wipe board or flip chart to collate feedback Computers with IDE installed
4	A1 Event triggers	IS	• **Individual activity:** Practice tasks to apply and reinforce knowledge of event triggers. • **Paired/small-group activity:** Learners to review programs within the groups. • **Plenary:** Summarise the lesson and use oral questioning to confirm learners' understanding.	Practice tasks Computers with IDE installed Dry wipe board or flip chart to collate feedback
5	A2 Controls and objects	IS	• **Teacher-led discussion/presentation:** How to use event controls and objects. • **Individual activity:** Give learners small extracts of code containing event controls and objects. They need to identify what is occurring in the programs. • **Plenary:** Discuss how learners coped with the activity and use oral questioning to identify any gaps in understanding.	Flip chart or similar for learners to record discussions and ideas Teacher presentation and notes Learner work from previous lesson Snippets of code

#	Topic	Lesson type	Suggested activities	Resources
6	A2 Controls and objects	IS	• **Teacher-led discussion:** Initiate a discussion about objects and controls within event-driven programming. • **Individual activity:** Learners use worksheets to create programs using objects and controls. • **Plenary:** Discuss how learners coped with the activity and use oral questioning to identify any gaps in understanding.	Flip chart or similar for learners to record discussions and ideas Learner work from previous lesson Computers with IDE installed Coding worksheets
7	A3 Event handlers	IS	• **Teacher-led presentation/discussion:** The purposes of, and how to use, common and custom event handlers. • **Paired/small-group activity:** Ask learners to research and present explanations (including extracts of code) on the purposes of common handlers and to demonstrate their effective usage. • **Plenary:** Discuss how learners coped with the activity and use oral questioning to identify any gaps in understanding.	Flip chart or similar for learners to record discussions and ideas Teacher presentation and notes

#	Topic	Lesson type	Suggested activities	Resources
8	A3 Event handlers	IS	• **Teacher-led discussion:** The need to develop/use event handlers. • **Individual activity:** Learners develop and design event handlers using a worksheet that specifies requirements that need to be developed • **Plenary:** Discuss how learners coped with the activity and use oral questioning to identify any gaps in understanding.	Flip chart or similar for learners to record discussions and ideas Computers with IDE installed Worksheet
9	A4 Handling data	IS	• **Teacher-led discussion:** The correct usage of basic identifiers. • **Paired/small group activity:** Learners work in their groups to examine the practice task materials and identify the most appropriate data types to use from available language set. • **Individual activity:** Develop simple applications using a variety of identifiers.	Flip chart or similar for learners to record discussions and ideas. Quiz Computers with IDE installed Practice task that outlines data required for different applications

#	Topic	Lesson type	Suggested activities	Resources
10	A5 Operators	IS	• **Teacher-led presentation:** The correct usage of operators. • **Paired/small group activity:** Divide the group into pairs or small groups and give each group a set of different scenarios that requires the use of operators. Groups are to develop these applications and present to the other groups. • **Learner presentation:** Each pair or small group presents their solutions to the class.	Flip chart or similar for learners to record discussions and ideas. Teacher presentation and notes Computers with IDE installed Scenarios to practise using operators
11	A7 Testing an event-driven program	IS	• **Teacher presentation:** Explain what testing involves for programs • **Individual activity:** Give learners a number of different scenarios and they have to develop and test a solution for each scenario.	Teacher presentation and notes. Computers with IDE installed Scenarios to practise developing and testing solutions

#	Topic	Lesson type	Suggested activities	Resources
12–14	Programming practice A1–A7	IS/AA	• **Lead-in:** Explain to learners that over the next three lessons they will complete a task that will allow them to practise all what they have learned so far in this unit. • **Individual:** Learners create a detailed solution for the program. • **Paired activity:** In pairs, learners should discuss their ideas/plans. They should be prepared to explain/justify their choices. Learners can provide peer feedback.	Computers with IDE installed Problem brief Test plan template
15–18	Learning aim A Practice assessment	IS/AA	• **Lead-in:** Explain to learners that over the next four lessons they will complete a practice assessment to prepare them for the final assessment of Learning aim A. • **Individual assessment activity:** Learners respond to a practice assessment activity like the final assessment. ***Note:*** *The assessment used should follow a similar structure to the final assessment and should be graded against the same criteria. However, the assessment* ***must not*** *be the same one that will be used to assess their final grade for the unit.*	Computers with IDE installed Practice assessment Test plan template

#	Topic	Lesson type	Suggested activities	Resources
19	Practice assessment feedback	AA	• **Lead-in:** Provide learners with the marked work and feedback on the outcomes from the practice assessment (lessons 15–18). • **Individual activity:** Learners read through the marked work and identify areas that need to be improved. Learners complete individual study on areas where they need improvement.	Computers with IDE installed Learner outcomes from lessons 15–18 Teacher feedback Online programming tutorial Practice programming exercises
20–24	Learning aim A Final assessment activity	IS/AA	• **Lead-in:** Introduce Task 1 of the assessment (see Assessment Workbook) highlighting the main assessment criteria, deadline and submission criteria. Highlight to learners that Unit 2 designs are needed for this assessment as they are developing the designs created in Unit 2. • **Individual assessment activity:** Using the Assessment Workbook, learners should complete Task 1.	Computers with IDE installed Unit 2 designs used in assessment Assessment workbook

#	Topic	Lesson type	Suggested activities	Resources
25	C2 Key features of object-oriented programming	IS	• **Teacher-led discussion:** To introduce the unit content, initiate a discussion on object-oriented programming, its characteristics and uses. • **Teacher presentation:** Give an overview of object-oriented programming and introduce the key features. • **Paired/small group activity**: Learners research the features of object-oriented programming. • **Plenary**: Summarise the lesson and use oral questioning to confirm learners' understanding.	Dry wipe board or flip chart to collate feedback. Teacher presentation and notes Computers with internet access
26	B1 Common coding terms and concepts of object-oriented programming	IS	• **Teacher presentation:** Explain the different terms and concepts of object-oriented programming. • **Paired/small group activity:** Learners research the different coding terms and concepts of object-oriented programming. • **Individual activity**: Look at code snippets of object-oriented programming to identify coding concepts**.**	Small dry wipe boards for paired activity/basic stationary for making notes Computers with internet access Teacher presentation and notes

#	Topic	Lesson type	Suggested activities	Resources
27	B2 Using an IDE for object-oriented programming	IS	• **Teacher presentation:** Give an overview of an object-oriented IDE and its features. • **Individual activity:** Learners use practice tasks to develop simple programs using the IDE. • **Plenary**: Summarise the lesson and use oral questioning to confirm learners' understanding.	Computers with IDE installed Teacher presentation and notes Practice tasks
28	B2 Variables and methods	IS	• **Teacher presentation:** Explain the variable types and methods used in object-oriented programming. • **Paired/small group activity:** Learners research the variables and methods used and produce a presentation on how each can be used. • **Plenary**: Summarise the lesson and use oral questioning to confirm learners' understanding.	Small dry wipe boards for paired activity/basic stationary for making notes. Computers with IDE installed Teacher presentation and notes
29	B2 Variables and methods	IS	• **Individual activity:** Learners create programs using different variables and methods, using worksheet activity sheets. • **Paired/small group activity:** Learners work in pairs or small groups to review each other's programs and provide feedback.	Computers with IDE installed Small dry wipe boards for paired activity/basic stationary for making notes

#	Topic	Lesson type	Suggested activities	Resources
30	B3 Syntax elements of object-oriented applications	IS	• **Teacher presentation:** Explain the syntax elements used in object-oriented programming. • **Paired/small group activity:** Learners research the syntax elements and produce a presentation on how each can be used. • **Plenary**: Summarise the lesson and use oral questioning to confirm learners' understanding.	Small dry wipe boards for paired activity/basic stationary for making notes Computers with IDE installed Teacher presentation and notes
31–33	B3 Syntax elements of object-oriented applications	IS	• **Individual activity:** Learners use practice tasks to create programs using different syntax elements. • **Paired/small group activity:** Learners work in pairs or small groups to review each other's programs and provide feedback.	Computers with IDE installed Small dry wipe boards for paired activity/basic stationary for making notes Practice tasks

#	Topic	Lesson type	Suggested activities	Resources
34–36	Programming practice B1–B4	IS	• **Lead-in:** Explain to learners that over the next three lessons they will complete a task that will allow them to practice all what they have learned so far in this unit. • **Individual:** Learners create a detailed solution for the program and test it. • **Paired activity:** In pairs, learners discuss their solutions. They should be prepared to explain/justify their choices. Learners can provide peer feedback.	Computers with IDE installed Small dry wipe boards for paired activity/basic stationary for making notes Programming scenarios
37–40	Learning aim B Practice assessment	IS/AA	• **Lead-in:** Explain to learners that over the next four lessons they will complete a practice assessment to prepare them for their final assessment of Learning aim B. • **Assessment activity:** Learners take a practice assessment activity that is like the final assessment. ***Note**: The assessment activity used should follow a similar structure to the final assessment and should be graded against the same criteria. However, the assessment used **must not** be the same one that will be used to assess their final grade for the unit.*	Computers with IDE installed Practice assessment activity Test plan template

#	Topic	Lesson type	Suggested activities	Resources
41	Learning aim B Practice assessment feedback	IS/AA	• **Lead-in:** Provide learners with the marked work and feedback on the outcomes from the practice assessment (lessons 37–40). • **Assessment activity:** Learners read through the marked work and identify areas that need to be improved. Learners complete individual study on areas where they need improvement.	Computers with IDE installed Learner outcomes lessons 37–40 with teacher feedback Practice programming exercises
42–46	Final assessment of Learning aim B	AA	• **Lead-in:** Introduce the assessment Task 2 (see Assessment Workbooks) highlighting the main assessment criteria, deadline and submission criteria. Highlight to learners that Unit 2 designs are needed for this assessment as they are developing the designs created in Unit 2. • **Individual assessment activity:** Using the Assessment Workbook, learners should complete Task 2.	Computers with IDE installed Assessment workbook Unit 2 designs used in assessment

#	Topic	Lesson type	Suggested activities	Resources
47	Learning aim C	IS	• **Teacher-led discussion:** To introduce the unit content, initiate a discussion on the key features of event-driven, object-oriented and procedural programming languages. Make use of materials from Unit 1, in which learners studied a procedural programming language. • **Teacher presentation:** Give an overview of the key features associated with each of the 3 programming languages. • **Plenary**: Summarise the lesson and use oral questioning to confirm learners' understanding.	Small dry wipe boards for paired activity/basic stationary for making notes. Computers with internet access Teacher presentation and notes Unit 1 specification for reference to procedural programming
48	Learning aim C	IS	• **Group activity:** Give each group a different programming language to research and create a presentation on how suitable the language is for a specified application. Each group to present findings.	Small dry wipe boards for paired activity/basic stationary for making notes Computers with internet access
49	Learning Aim C Final assessment	AA	• **Lead-in:** Introduce the assessment Task 3 (see Assessment Workbooks) highlighting the main assessment criteria, deadline and submission criteria. • **Assessment:** Using the Assessment Workbook, learners should complete Task 3.	Assessment Workbook

#	Topic	Lesson type	Suggested activities	Resources
50	Unit evaluation	IS	• **Teacher-led discussion:** To recap the unit content, initiate a discussion on the final assessments. • **Individual activity:** Learners complete a feedback questionnaire of the unit identifying improvements.	Small dry wipe boards for activity/basic stationary for making notes Feedback questionnaire

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	1 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the key features of event-driven programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • **PS** Presentation on unit introduction • **PS** Presentation on key features of event-driven programming • Case studies • Event-driven programming summary: youtu.be/qrA7eD18CZo • Concepts of event–driven programming ('Polling' vs. EDP) in Python youtu.be/q8O7zgh59P4 • Key characteristics of event–driven programming: youtu.be/qrA7eD18CZo
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Introduction to unit (30 minutes)	• **Teacher:** Introduction to unit. Use oral questioning to find out about learners' experience of event-driven programming.
Main activity – Introduction to key features of event-driven programming (130 minutes)	• **Teacher presentation**: Overview of the features of event-driven programming. • **Individual learner activity:** Learners use case studies to understand why event-driven languages are required.
Concluding activity (20 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	2 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • be able to use an integrated development environment for event-driven programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • **PS** Presentation on IDE for event-driven programming • **AS** simple programs • Introduction to IDEs: youtu.be/o–a1uXkcNXY • Introduction to the Android IDE: youtu.be/K2dodTXARqc • The VBA Integrated Development Environment: youtu.be/CHL21XbkYOU
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (30 minutes)	• **Teacher-led discussion**: Introduce IDEs and ask learners about their experience using different IDEs. • **Teacher presentation**: Identifying the need for an IDE and the benefits of using it.
Main activities – using an IDE (140 minutes)	• **Individual activity**: Learners use activity sheets to develop simple programs using the IDE.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	3 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand event triggers and the need for them • be able to code event triggers.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • Basic explanation of events, event triggers and event loops: youtu.be/t4YnoVVlYIU • Triggering events and events triggered through user interaction: informit.com/library/content.aspx?b=STY_Csharp_24hoursand seqNum=50 • Explanation of event triggers: kab13098832.weebly.com/trigger–functions.html
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher-led discussion:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Paired/small-group activity:** Learners should research using the internet, and document different event triggers used in event-driven programming. • **Paired/small-group activity:** Learners should create code that uses event triggers.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	4 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • be able to identify code event triggers in event-driven programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • Learner practice tasks on event triggers
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Individual activity**: Give learners several activities that allow them to apply code for different event triggers. • **Paired/small-group activity:** Learners should review their coding activities with others.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	5 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the purposes of, and how to use, event controls and objects.

Resources checklist	• Dry wipe board or flip chart to collate feedback • **PS** Presentation on how use of controls and objects in event-driven programming • Learner activity material – code snippets • Coding controls and objects in Visual Basic: youtu.be/n_1Bq–JcGmQ • Declaring Objects, Setting Objects, Passing Control Arrays And Objects in Visual Basic 6: youtu.be/ZT4CZZCsMXg • Explanation of programming event controls and objects in VB.Net: teacherialspoint.com/vb.net/vb.net_basic_controls.htm
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Teacher presentation/discussion**: Discuss how to use event controls and objects within event-driven programming. • **Individual activity**: Give learners code snippets. They need to identify what is occurring in each program.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	6 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • be able to use objects and controls to develop programs in event-driven programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • Learner work from previous lesson • **AS** Learner activity material – program scenarios learners must develop
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Teacher discussion**: Discuss how objects and controls are used in event-driven programming. • **Individual activity**: Learners develop programs against given scenarios in worksheets making use of objects and controls.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	7 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand how to use common and custom event handlers • understand the purposes of common and custom event handlers.

Resources checklist	• Dry wipe board or flip chart to collate feedback • **PS** Purposes of common and custom event handlers • Definition of event handlers: searchsoa.techtarget.com/definition/event–handler
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Teacher presentation/discussion**: Discuss the purposes of, and how to, use common and custom event handlers. • **Paired/small group activity**: Learners research and present explanations on the purposes of common and custom event handlers to demonstrate their usage.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	8 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • be able to create code for event handlers in event-driven programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • **AS** Event handler activity worksheet • Small dry wipe boards for paired activity/basic stationary for notetaking • Computers with IDE installed
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Teacher-led discussion:** Discuss the need to develop/use event handlers**.** • **Individual activity:** Learners develop and design event handlers to meet specifications of given requirements provided in activity worksheet. • **Teacher-led discussion:** Discuss what learners have developed and go through the different event handlers
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	9 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • be able to use basic identifiers in event-driven programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • Quiz • Practice task that outlines data required for different applications
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (30 minutes)	• **Teacher:** Recap previous lesson and introduce aims and objectives for this lesson. • **Learner activity:** Give learners a quiz to recap what was covered in the previous lesson. This can be self- or peer-marked.
Main activities (140 minutes)	• **Teacher-led discussion:** The correct usage of basic identifiers. • **Paired/small group activity:** Learners work in their groups to examine real-world data and identify the most appropriate data types to use from available language set. • **Individual activity:** Learners develop simple programs using a variety of identifiers.
Concluding activity (10 minutes)	• **Plenary**: Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	10 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • be able to develop code that uses operators.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • **PS** Teacher presentation and notes. • **AS** Operators scenario • Small dry wipe boards for paired activity/basic stationary for notetaking
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Teacher-led presentation:** The correct usage of operators. • **Paired/small group activity**: Learners develop code for a number of different scenarios that requires the use of operators. • **Group presentations**: Groups present their developed code to other groups, justifying reasons why different operators were used.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	11 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand what testing is involved when developing program code • be able to develop and test small applications.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • **AS** scenarios for practical activities • **PS** How to test a program • How to create a test plan: youtube/46KB0-yrvvQ • What is a test plan?: youtube/Vdm1Lh540LM • How to Write a Test Plan Document from Scratch: softwaretestinghelp.com/how-to-write-test-plan-document-software-testing-training-day3/ • How to Write a Great Software Test Plan: microtoolsinc.com/howstp
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Introduction (20 minutes)	• **Teacher:** Recap previous lesson and introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Teacher presentation:** Explain what testing involves for programs. • **Individual activity:** Give learners scenarios that need to be developed and tested. • **Teacher discussion:** Discuss why different testing methods have been used.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	12–14 (9 hours)

Lesson objectives	At the end of the three lessons, learners will: • be able to create functional programs in event-driven programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • **AS** problem brief with program scenarios
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (10 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (8 hours 40 minutes)	• **Teacher discussion**: Give an overview of the given problems that require a developed solution. • **Individual activity**: Learners should create a detailed solution for the outlined problem. • **Paired activity:** Learners should discuss their solutions. They should be prepared to explain/justify their choices. Learners can provide peer feedback.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	15–18 (12 hours)

Lesson objectives	At the end of the four lessons, learners will: • be able to complete a practice assessment on event-driven programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • Practice assessment.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Explain to learners that over the next four lessons they will complete a practice assessment to prepare them for their final assessment.
Main activities (11 hours, 40 minutes)	• **Individual activity**: Learners complete a practice assessment brief like the final assessment.
Assessment: completed outside of class time	• *Teacher: Collect learners' work from lessons 15–18. Identify areas of strength and areas of improvement within written feedback.*

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	19 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the requirements of the assessment for Learning aim A.

Resources checklist	• Computers with IDE installed • Learner outcomes from lessons 15–18 • Teacher feedback • Practice programming exercises
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Introduce aims and objectives for this lesson and provide learners with assessed work.
Main activities (150 minutes)	• **Teacher presentation**: Give an overview of the requirements of the practice assessment and identify areas done well and areas that need to be improved. • **Individual activity**: Learners should read through the marked work and identify areas that need to be improved. They then complete additional practice programming exercises.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	20–24 (15 hours)

Lesson objectives	At the end of the five lessons, learners will: • be able to carry out the work required for the final assessment of Learning aim A.

Resources checklist	• Computers with IDE installed • Assessment Workbook • Learners' designs created for the Unit 2 assessment
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (10 minutes)	• **Teacher:** Recap previous lesson and introduce the assessment. Remind learners that their programs for Unit 4 will need to make use the designs they created for the Unit 2 assessment.
Main activities (14 hours, 40 minutes)	• **Individual assessment activity:** Learners complete Task 1, using their Assessment Workbook.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the process for assessment and dates when learners can expect feedback.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	25 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the key features of object-oriented programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • **PS** Teacher presentation and notes. • Computers with internet access • Video giving an introduction to object-oriented programming: youtube.com/watch?v=NUl8lcbeN2Y
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher**: Recap previous lesson and introduce aims and objectives for this lesson. Use oral questioning to confirm learners' knowledge of object-oriented programming.
Main activities (150 minutes)	• **Teacher presentation**: Explain the key features of object-oriented programming • **Paired/small group activity**: Learners research the features of object-oriented programming.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	26 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand common coding terms and concepts of object-oriented programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • **PS** Teacher presentation and notes • Computers with internet access • An introduction to the many concepts within object-oriented programming (including objects): youtube/yu8-u-MdW iU • Modelling real world items using classes and objects: youtube/tWIe9E4SWQo • Adobe article on the concepts of objects and classes within object-oriented programming: adobe.com/devnet/actionscript/learning/oop-concepts/objects-and-classes.html
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Introduction (20 minutes)	• **Teacher**: Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Teacher presentation**: Explain common coding terms and concepts of object-oriented programming. • **Paired/small group activity**: Learners research common coding terms and concepts. • **Individual activity**: Look at code snippets of object-oriented programming to identify coding concepts.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	27 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • be able to use an IDE for object-oriented programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • **PS** Teacher presentation and notes • **AS** Activity sheet of practice tasks for learners to develop simple programs
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Teacher presentation**: Demonstrate an IDE that is used for object-oriented programming • **Individual activity**: Learners use practice task activity sheets to develop simple programs using the IDE.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	28 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the need for variables and methods in object-oriented programming • be able to code with variables and methods in object-oriented programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • **PS** Teacher presentation and notes • Video looking at Classes, Objects, Data and Operations, Instance Variables, Class Variables, Instance Methods, Class Methods, Constructors: youtube/rDro2FzfMR8 • Web page explaining objects, instance methods, and instance variables: math.hws.edu/javanotes/c5/s1.html
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	Teaching notes
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Teacher presentation**: Explain the variable types and methods used in object-oriented programming. • **Paired/small group activity**: Learners work in pairs or small groups to produce a five to ten-minute presentation on how each method and variable type can be used in object-oriented programming and present to whole class.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	29 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the need for variables and methods in object-oriented programming • be able to code with variables and methods in object-oriented programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Individual activity:** Learners create functional programs in response to a worksheet containing several scenarios. • **Paired/small group activity:** Review each other's programs and provide feedback.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	30 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand how syntax elements are used in object-oriented programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • **PS** Syntax elements presentation
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Teacher presentation**: Explain the syntax elements used in object-oriented programming. • **Paired/small group activity**: Learners work in pairs or small groups to produce 5–10-minute presentation on how each syntax element can be used in object-oriented programming and present to whole class.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	31–33 (9 hours)

Lesson objectives	At the end of the three lessons, learners will: • be able to create functional programs using different syntax elements in object-oriented programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • **AS**: Practice tasks
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (8 hours 30 minutes)	• **Individual activity:** Give learners a worksheet containing several scenarios that include syntax elements. Learners they have to create a functional program for each scenario. • **Paired/small group activity:** Review each other's programs and provide feedback.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and uses oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	34–36 (9 hours)

Lesson objectives	At the end of the three lessons, learners will: • be able to create functional programs in object-oriented programming.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • **PS** Teacher presentation and notes • **AS** program scenarios.
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (8 hours 30 minutes)	• **Teacher discussion:** Give an overview of the given problems that require a developed solution. Focus on the key areas of development that must be incorporated: objects, classes and loops. • **Individual activity:** Learners should create a detailed solution for the outlined problem. They should develop and test the solution to ensure it is functional. • **Paired activity**: Learners should discuss their solutions. They should be prepared to explain/justify their choices. Learners can provide peer feedback.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and uses oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	37–40 (12 hours)

Lesson objectives	At the end of the four lessons, learners will: • be able to complete a practice assessment for Learning aim B.

Resources checklist	• Practice assessment activity • Computers with IDE installed • Test plan template
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Explain to learners that over the next four lessons they will complete a practice assessment to prepare them for their final assessment.
Main activities (11 hours 40 minutes)	• **Individual activity**: Learners complete a practice assessment brief like the final assessment.
Assessment: completed outside of class time	• ***Teacher:*** *Collect learners' work from lessons 37–40. Identify areas of strength and areas of improvement within written feedback.*

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	41 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • be prepared to take the assessment for Learning aim B.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with IDE installed • Learner outcomes from lessons 37–40 with teacher feedback
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Introduce aims and objectives for this lesson and provide learners with assessed work**.**
Main activities (150 minutes)	• **Teacher presentation:** Give an overview of the requirements of the practice assessment and identify things done well and areas that need to be improved. • **Individual activity:** Learners should read through the marked work and identify areas of improvement and complete individual study on these areas.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	42–46 (15 hours)

Lesson objectives	At the end of the five lessons, learners will: • be able to carry out the work required for the final assessment of Learning aim B.

Resources checklist	• Computers with IDE installed • Assessment Workbook • Learners' designs created for the Unit 2 assessment
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (10 minutes)	• **Teacher**: Recap previous lesson and introduce the assessment. Remind learners that their programs for Unit 4 will need to make use the designs they created for the Unit 2 assessment.
Main activities (14 hours, 40 minutes)	• **Individual assessment activity:** Learners complete Task 2 using their Assessment Workbook**.**
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the process for assessment and dates when learners can expect feedback.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	47 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the key features of event-driven, object-oriented and procedural languages.

Resources checklist	• Dry wipe board or flip chart to collate feedback • **PS** Overview of key features • The specification for Unit 1 for reference to procedural programming • Computers with internet access
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Teacher led discussion:** Discuss the key features of each programming language and assess how much knowledge learners have. • **Teacher presentation:** Discuss the key features of each programming language • **Individual activity:** Learners to look back over their developed programs (including programs developed in *Unit 1: Introduction to Programming*) and identify key features for each programming language.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	48 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • understand the key features of event-driven, object-oriented and procedural languages.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Computers with internet access
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (20 minutes)	• **Teacher:** Recap previous lesson with oral questioning to confirm understanding. Introduce aims and objectives for this lesson.
Main activities (150 minutes)	• **Group activity:** Split learners into groups and give each group a programming language. They are to research and create a 15-minute presentation on that language and how suitable it is for a specified application. Groups to present their findings to whole class. • **Teacher-led discussion:** Discuss the findings of each programming language.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the main learning points identified in the lesson and use oral questioning to confirm learners' understanding.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	49 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • be able to carry out the work required for the final assessment of Learning aim C.

Resources checklist	• Assessment Workbook
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (10 minutes)	• **Teacher:** Recap previous lesson and introduce aims and objectives for this lesson.
Main activities (160 minutes)	• **Individual activity**: Learners complete Task 3 in their Assessment Workbook.
Concluding activity (10 minutes)	• **Plenary:** Teacher confirms the process for assessment and dates when learners can expect feedback.

Lesson plan

Qualification	Pearson BTEC Uzbekistan Level 4 Qualifications in Software Development
Unit	Unit 4: Programming Using Different Coding Paradigms
Lesson number	50 (3 hours)

Lesson objectives	At the end of the lesson, learners will: • know how well they did in the assessment • express their views on the unit.

Resources checklist	• Dry wipe board or flip chart to collate feedback • Feedback questionnaire
Key: **AS**: Activity Sheet; **TF**: Template Form; **PS**: Presentation Slide	

Activities	**Teaching notes**
Introduction (10 minutes)	• **Teacher:** Introduce aims and objectives for this lesson.
Main activities (170 minutes)	• **Teacher-led discussion**: Discuss the whole unit and gather feedback on how it was delivered and assessed. • **Individual activity**: Complete unit evaluation and identify improvements that could be made.

Rr301219 ENG_9780997883039_UZBEK_TRC1BK1_SOFTWARE_UNIT4_RR.DOCX.1-62/0

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

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

VAT Reg No GB 278 537121

ISBN 978-0-9978830-3-9